adafruit learning system

Cupcade: the Raspberry Pi Micro Arcade Cabinet

Created by Phillip Burgess

Last updated on 2015-09-15 02:00:19 PM EDT

Guide Contents

Guide Contents 2
Overview 4
Kit Contents: 4
Optional additions: 5
Plan Ahead 7
PiTFT Assembly & Test 10
Your Cupcade SD card 10
Solder Time! 11
If there's no response… 13
Success! 14
Now a Second Test… 15
Test ROM 16
Interface Board 17
Installing ROMs 22
Downloading the Free ROMs 22
Installing ROMs 24
Dry Run 27
If the Pi Does Not Boot 31
If the Pi Boots Successfully 31
When You're Done Testing 32
Cabinet Part 1 33
Cabinet Part 2 41
Joystick and Buttons 42
Screen and Audio 44
Cabinet Part 3 45
Okay, the Tricky Part 47
Cabinet Part 4 53
If the system does not boot… 59
NES Emulation 61
NES Controls 61
Details and Errata 64

System Setup 64
Burning your own SD card 64
Core OS and TFT Support 64
AdvanceMAME 66
Retrogame and Gamera 67
Custom Cabinet Art 68
Troubleshooting! 70
Known Issues 71
Fixing Sound Samples 71
Beta Board Part 1 73
Beta Board Part 2 84
Arts & Crafts Time! 88
Beta Board Part 3 90
Button Wires 90
Joystick Wires 94
Power Wires 95
Audio Wires 97
Soldering’s done! 99

Overview

A past Maker Faire display of mine once incorporated a *Pac-Man* theme “for the old-timers.” It was a surprise then to see young kids all recognized the characters too. How? Smartphones! Thanks to *emulation* — running old code byte-for-byte on modern hardware — these classic games are still played and relevant a generation later.

Much of the mystique of the originals lied in the cabinets and controls. *Anyone* can load a game on a smartphone or tablet…but the *physicality* of the arcade machine and its clicky buttons made them rare objects of desire back in the day. We wanted to capture a small taste of that, using the tiny Raspberry Pi computer. The result is a DIY kit we call **Cupcade!**

Cupcade isn’t the first, but it’s notable for using the Adafruit PiTFT display. The direct digital interface delivers a pixel-perfect rendition of classic games with none of the blurriness you’d get with a composite screen.

Kit Contents:

- Adafruit PiTFT Mini Kit (http://adafru.it/1601)

- Analog joystick with breakout board (http://adafru.it/512)
- 16mm Buttons (http://adafru.it/1445) (4)
- Arcade Button Quick Connect Wires (http://adafru.it/1152) (4)
- Female/female jumper wires (http://adafru.it/266) (strip of 4)
- *Cupcade* Raspberry Pi interface board and related components
- 26-pin ribbon cable
- 8 Ohm 1 Watt speaker (http://adafru.it/1313)
- Right-angle 3.5mm audio plug to pigtail cable (http://adafru.it/1700)
- Acrylic case parts (15)
- 4-40 and 2-56 screws and nuts (15 and 3 ea., respectively)
- Nylon board standoffs (2)
- Piece of heat-shrink tubing
- Blank 4GB SD card
- USB charger/power supply (http://adafru.it/dfY)
- USB A to micro B cable (http://adafru.it/1513)

Earlier "beta" kits had some additional parts not listed here; new kits have these pre-assembled on the interface board.

You will also need:

- Raspberry Pi Model B computer (http://adafru.it/998) (Model A (http://adafru.it/1344) also works, but with only half the RAM this may impact performance). *If you have a Model B on-hand, the earliest 'V1' boards will not work with this project (easy to spot — they have no mounting holes).* ***The new Model B+ is not currently compatible, but will likely be addressed in the future.***
- Soldering iron, solder, stranded or solid core wire (24 or 22 gauge) and related paraphernalia
- Masking tape
- For setup, you may temporarily need a keyboard and monitor
- Game ROM files

Optional additions:

- USB Miniature WiFi Module (http://adafru.it/814)
- Wireless keyboard (http://adafru.it/1737)
- USB Battery Pack (http://adafru.it/1566) for portable use

> Cupcade requires a Raspberry Pi Model B with mounting holes. The new B+ board, and early Model Bs with no mounting holes, will NOT work.

Hey! Don't be fooled by the fun-and-games nature of this project. It's a challenging build that draws on a broad range of maker skills: fiddling with Linux commands, stripping and soldering wires, and even a bit of arts & crafts. Read through the whole guide before starting, decide if you're ready to

tackle this and make sure you have everything you need.

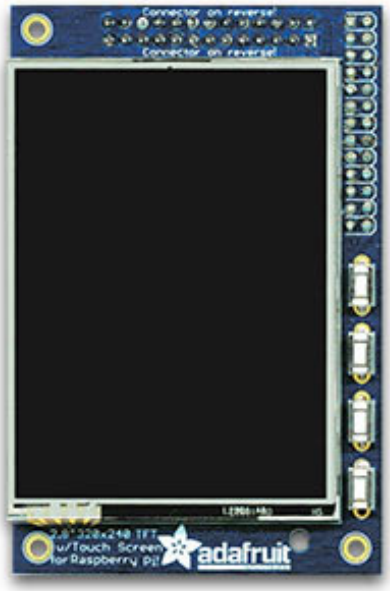
Our original **Retro Gaming with Raspberry Pi** (http://adafru.it/dcs) guide is a little easier — same goal, fewer pieces, using a regular computer monitor for the display. You might want to start there if this project looks a bit overwhelming.

Plan Ahead

This is a red pill / blue pill moment.

Think a bit about the “old school” games you like to play. Home console games — those that connect to your TV — have a horizontal screen orientation. But in arcades, many games took advantage of a *vertical* screen.

Cupcade can use either a **horizontal** or **vertical** screen orientation, but this must be decided early on. It *can* be changed later, but this is a non-trivial operation; you’ll need to dismantle and reassemble the whole thing.



Horizontal Screen

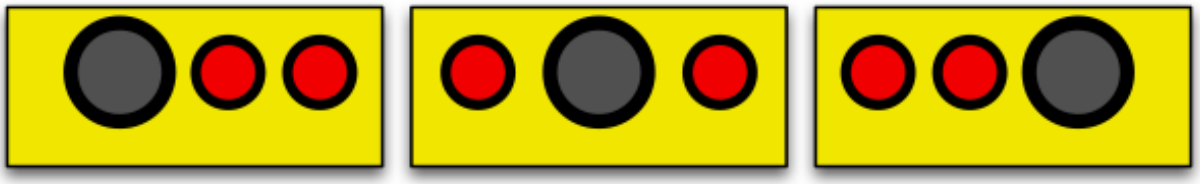
- Pole Position
- Zero Wing
- Most Fighting Games

Vertical Screen

- Pac Man
- Dig Dug
- Zaxxon

NES (Nintendo) games (more on this later) *require* a horizontal screen; that emulator won't work vertically. Arcade games give you a choice, depending on the ROM

There are also **three different ways** the controls can be arranged…but again, this is a commitment and a fair bit of work to change later.



Joystick Left

Best for 2-button games.

- Scramble
- Qix
- Xevious

Joystick Center

Best for single-button or buttonless games.

- Pac Man
- Galaga
- Joust

Equally usable by right- or left-handed players.

Joystick Right

Alternate 2-button layout for lefties.

Therefore: **think ahead** about which game(s) you're most likely to play. Test them out on the Raspberry Pi *before* assembling the whole case around it. Determine which perform acceptably with the emulator and look good on the screen…then choose a display and control combination that works well for the most (or most desirable) games from your list. Others can still be played, they're just less than optimal.

This basic joystick-and-two-buttons combo covers a vast number of classic arcade games. You can optionally plug in a USB keyboard for anything more complex.

PiTFT Assembly & Test

Given this project's complexity, we'll be testing and re-testing the system to validate our progress. Confirm your system **passes each test** before advancing to the next. Mis-steps are very time-consuming!

We'll build and test the Raspberry Pi and PiTFT display with a USB keyboard plugged in. The arcade controls and cabinet will come later.

Your Cupcade SD card

Because your kit comes with a **blank** SD card, and needs to be **burned** with the Cupcade image

Begin by downloading our pre-built Linux disk image for the Cupcade. This saves a *whole lot* of configuration steps!

Download Cupcade Image 5-13-2014

http://adafru.it/ds6

This is a big file (about 840 megabytes) and will take a while to transfer.

After downloading, the disk image needs to be installed on an SD card (4GB or larger). If you're new to this, the process is explained in **Adafruit's Raspberry Pi Lesson 1** (http://adafru.it/aWq) and/or check out this tutorial on how to burn SD cards with Raspbian images (http://adafru.it/aMW)

The disk image contains the operating system and ROM selector BUT **no games are included**. (http://adafru.it/dfZ)**You will need to track down your own ROM files to complete the process.** - check the Installing ROMs step for more deails Google can help.

We designed this card so most users can simply drag-and-drop ROMs using any computer with an SD card reader. For most people (people who just want to turn on the machine and play), an external monitor or WiFi is not required.

We do suggest connecting any USB keyboard to help while testing the PiTFT and to set up your screen orientation.

WiFi is optional and up to you. More technical users might find it helpful for transferring ROM files or for remote login via ssh to configure the system…but it's not a necessity and most users will do fine plugging the SD card into their regular computer to add game files, or a keyboard for configuration. If it's desired, plug in a USB mini WiFi adapter now and set up wireless networking following the first boot test.

Solder Time!

It's time to build your kit! Heat up your soldering iron and lets get started!

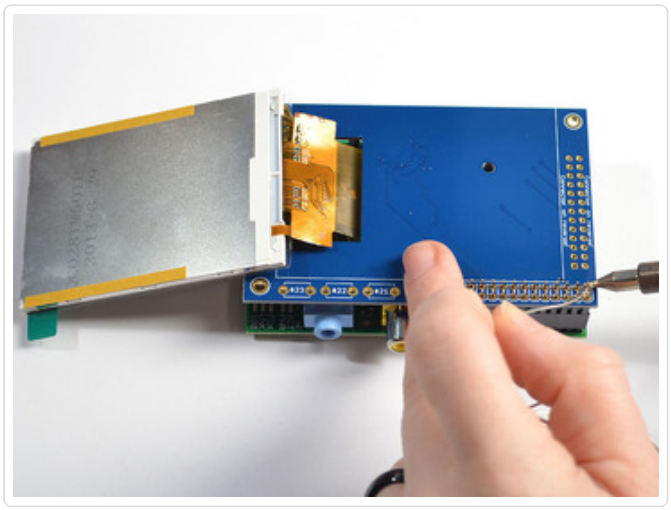
The PiTFT may come pre-assembled. If so you can skip the soldering part here

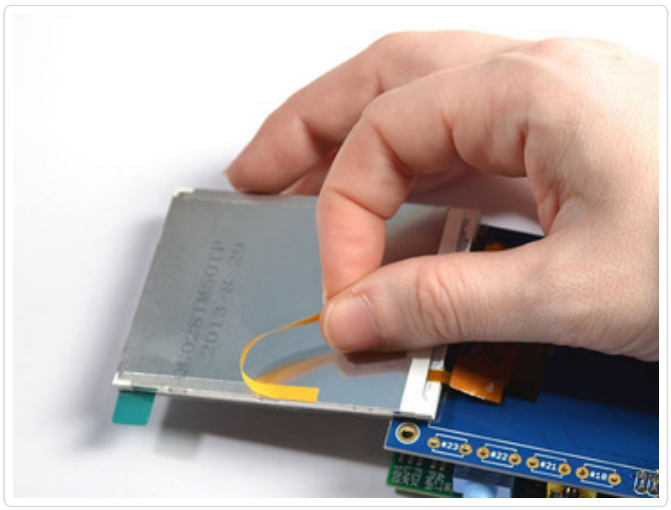
The tall female (socket) and shorter male (pin) headers will be installed in the positions shown. The female socket goes on the long edge of the PiTFT board, male on the short edge (labeled “Connector on reverse!” on the silkscreen side of the board).

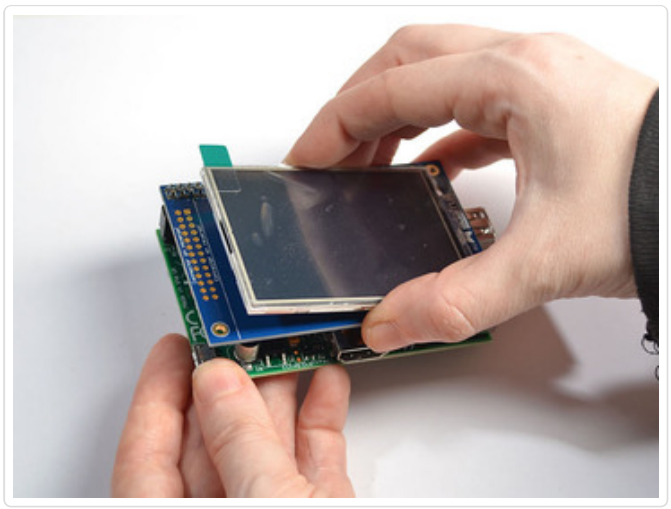
Note the position of the key (notch) on the male header, facing the center (not outside edge) of the board. Vitally important!

When soldering the female header, the Raspberry Pi can be used as a “stand” to hold the header and board in alignment. For the male header, you’ll probably need to hold it in place temporarily with masking tape.

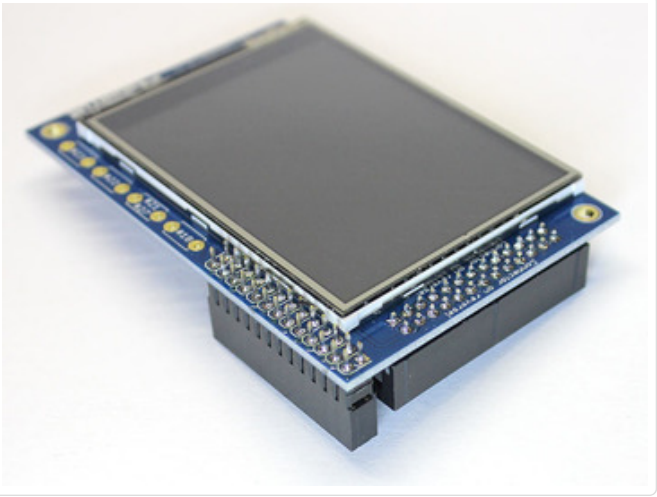
After soldering, peel the backing off the tape strips and position the screen on the board, leaving a little space between the header and the metal edge of the screen.







This is what good soldering looks like. The solder flows smoothly between the pins and the pads on the board, no blobs, bridges or gaps. **The Adafruit**



Guide to Excellent Soldering (http://adafru.it/cA6) has pointers for novices.

Once you're satisfied with the soldering and screen alignment, the protective film can be peeled from the display. This is not a screen protector like on your phone!

Install the PiTFT on the Raspberry Pi, making sure the header pins are aligned.

Connect a USB keyboard (not shown here) and plug the Raspberry Pi into a power supply.

Within about 10 seconds of connecting power, you should see the display come to life, with geeky Linux boot messages scrolling by, and eventually a game selection menu (list will be empty if no games have been loaded on the card yet).

If there's no response…

If nothing happens for 30 seconds or longer…either a black or white screen…disconnect power, **remove the PiTFT board,** plug the Raspberry Pi into a **regular monitor** and power up again.

You'll probably get a few lines of text and then it will stop; the Pi tries to switch over to the PiTFT at this point, but it's not present now. Press ALT+F2 on the keyboard and it should switch back to the desktop monitor with a login prompt.

If you get a login prompt on the monitor after pressing ALT+F2: Pi and SD card are fine, PiTFT is not. Examine your soldering closely, looking for any bridges or solder balls. Make sure the headers on the PiTFT and Raspberry Pi board are correctly aligned. Make sure the metal edge of the LCD is not contacting any of the header pins.

If it does not boot on either the PiTFT or monitor: possibly something wrong with the SD card; perhaps the operating system was not properly installed or something is amiss with the PiTFT assembly. Is it plugged in right? (Sometimes the header pins are off-by-one and it doesn't work.)

Do not proceed until you have a working system, including the PiTFT.

Success!

First time you boot from this card, the system will go through an automated reboot cycle once as the filesystem is expanded to use the full SD card. Second time around, it should boot all the way to the game menu…that's great and we'll prepare for the next test.

If the **screen orientation is correct** and you're **not adding WiFi:** press the ESC key on your keyboard. The Pi will initiate an orderly shutdown. Wait for the "Will now halt" message before disconnecting power.

If you need to **change the screen orientation:** press SHIFT+R while the game menu is displayed. The system will reboot using the new layout. If you need to change this again later, you'll need a USB keyboard connected; it's not possible from the game controls alone.

If you are **adding WiFi:** you'll need to log in and enter Linux commands. Press ALT+F3 for a login prompt, entering "pi" and "raspberry" for username and password. Then we have a guide for configuring wireless networking (http://adafru.it/cg7). You can snoop around and do any other system configuration you'd like (hostname, password, etc.). When you're done, either type sudo shutdown -h now or press ALT+F1 to return to the game menu and press the ESC key. Wait for the "Will now halt" message before disconnecting power.

If the game menu is empty: no ROM files have been loaded yet. Shut down the system (press the ESC key), insert the card in an SD reader on your regular computer, and place one or more arcade ROM files in the (http://adafru.it/dfZ)advmame/rom (http://adafru.it/dfZ) folder on the disk called (http://adafru.it/dfZ)boot (http://adafru.it/dfZ). (http://adafru.it/dfZ)

> Never just pull the plug on a Linux system, they hate that. Always use the shutdown command by pressing the ESC key or you risk losing data on the SD card.

Now a Second Test…

With the system powered off, remove the PiTFT board from the Raspberry Pi. Take the ribbon cable

included with your kit…

The header on the PiTFT is “keyed” so that pin 1 (the white wire) is always on the correct side. However, there’s two ends to the cable. Plug in the end that places the cable running behind the PiTFT board, not out in the open.

Careful now…the header on the Raspberry Pi board is *not* keyed! Pin 1 (white wire) should be near the edge of the board with the SD socket. Make sure the pins are aligned, not off by one. This end of the cable *should* point out in the open, not overlapping the Pi.

Turn the screen over and connect a USB keyboard and power. After a few seconds you should see the same bootup sequence as before.

If the PiTFT worked **when connected directly to the Pi** but **does not work with the ribbon cable:**

- Double-check the cable orientation and alignment, make sure nothing’s turned around or off by one.
- Shut the system down, remove the ribbon cable and re-connect the display directly atop the Pi. If it won’t boot now (but did before), something’s gone wrong with the SD card. Otherwise…
- If everything checks out but it still won’t boot with the ribbon cable connected, the cable may be defective. Visit the Adafruit Customer Support Forums, post a photo of your hardware and we’ll look it over for any gremlins or will have a replacement sent.

Test ROM

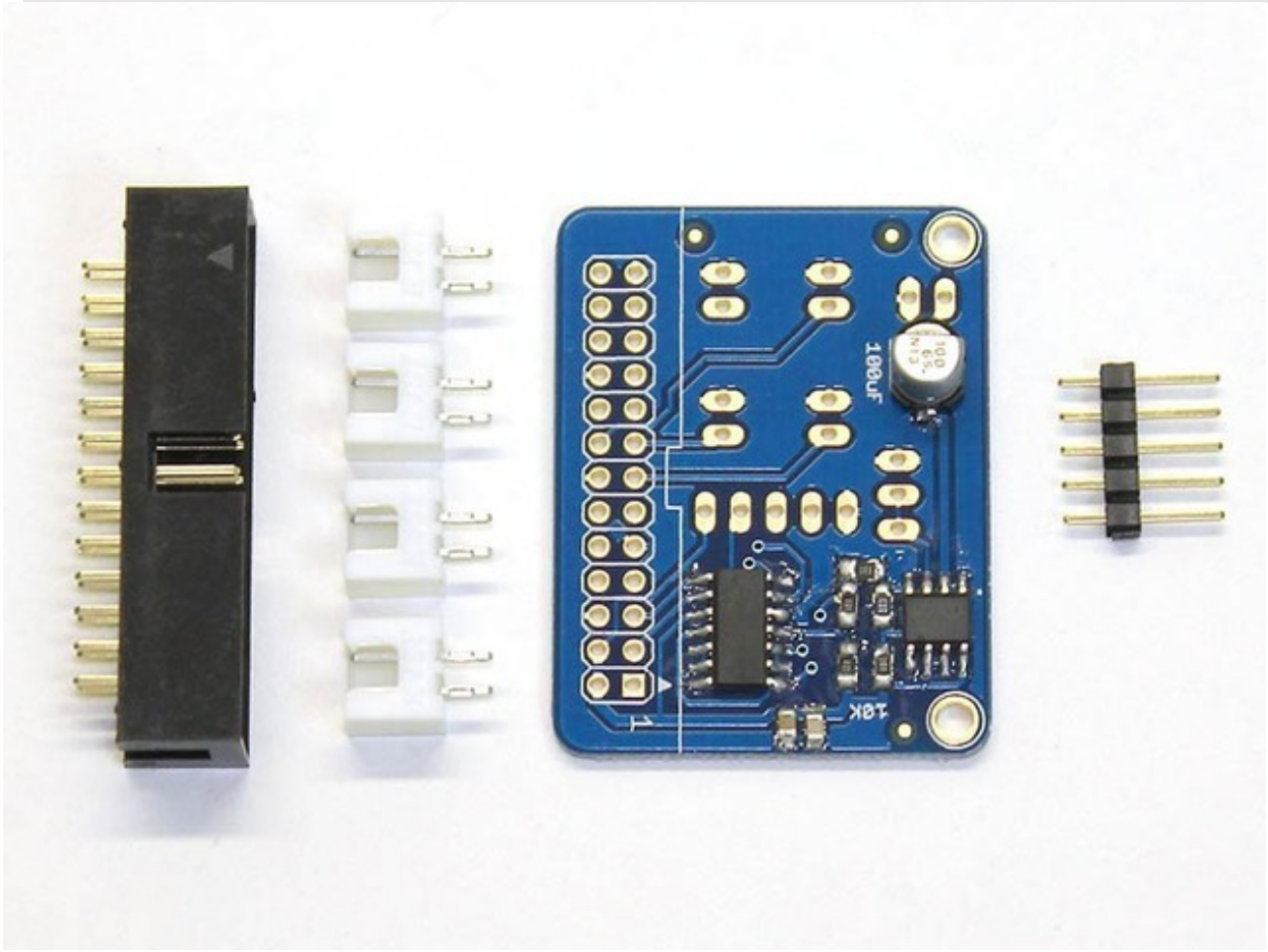
If you have any ROM files loaded in the boot/advmame/rom folder, you can select a game from the menu (using the arrow keys and enter) and see if it works. This is a good time to decide which games run well or not.

Press the ESC key to exit and return to the game menu.

This concludes the first phase of testing. Do a proper shutdown again (ESC from the game menu) and wait for the “halt” message before disconnecting power.

Do not continue to the next step until you have a working system.

Interface Board



A small circuit board — specially designed just for Cupcade — features a joystick adapter circuit and a small audio amplifier. Just a few through-hole components need to be soldered.

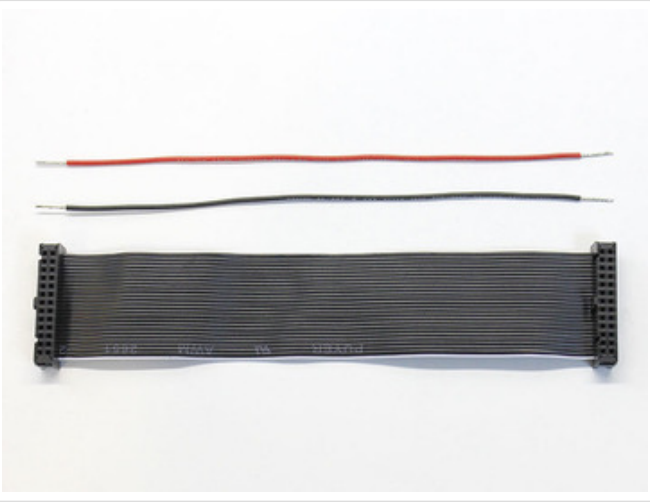
BETA USERS: the first version of this kit had a D.I.Y. interface board. Those pages have been moved to the end of the guide.

Different parts will get soldered on opposite sides of this board. Labels and outlines (along with the photos here) should make it clear where parts are inserted.

But first…a few wires need to be cut to about six inches long. You don't need a ruler for this…the ribbon cable is just the right length. Unplug it from the Pi and PiTFT and use it for reference.

This isn't rocket surgery, don't worry if wires aren't precisely the right length. Six-*ish* inches is fine!

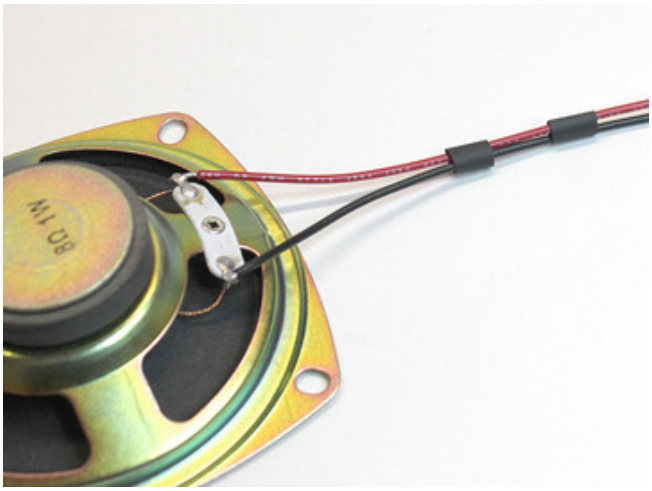
Cut two (2) wires about six inches long and strip



about 1/4" of insulation from both ends.

24 gauge stranded wire is ideal. A little thicker or thinner is okay, as is solid-core wire…stranded is simply more flexible.

Color-coding the wires likewise isn't essential; one color will suffice if that's what you have.



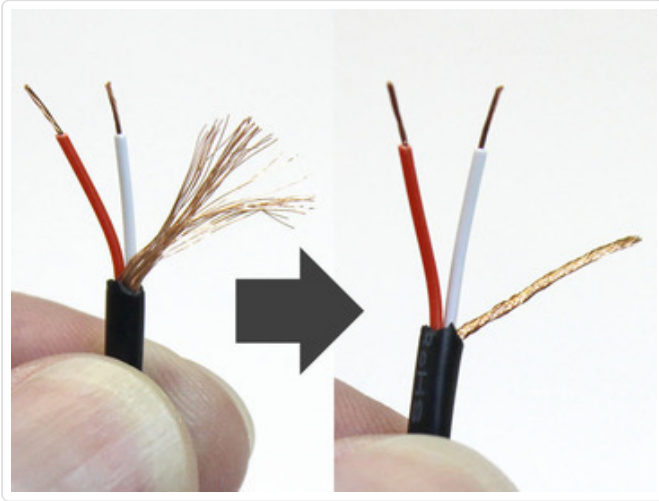
Solder one end of these wires to the connection points on the speaker.

Optional: you can keep these wires together with a bit of heat-shrink tube if you like.

Cut the audio cable so it's a little longer than the ribbon cable.

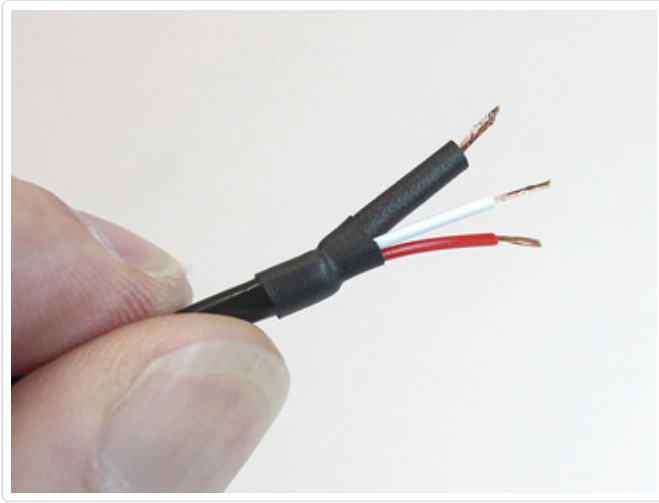
Strip about 1" of the *outer jacket only* to reveal the three wires inside: two are insulated, the third is bare copper.

Strip 1/4" insulation from the two inner wires. Twist



the stranded copper wire into a tidy bundle.

Some headphone cables have red and black wires rather than the red and white shown here. This is OK, it all works the same. Red is the right channel, black or white is the left channel, copper is ground.



That bare copper wire on the audio cable is a problem…it might contact metal parts in the vicinity. **We must insulate!**

Slide a small piece of heat-shrink tube a few inches down the cable (around the whole thing, not just the copper wire).

Slide a second piece of heat-shrink tube over just the copper wire and heat it up. This covers most of the wire, but there's still a tiny gap.

Slide the first (un-shrunk) piece back up so it covers the gap, apply heat.

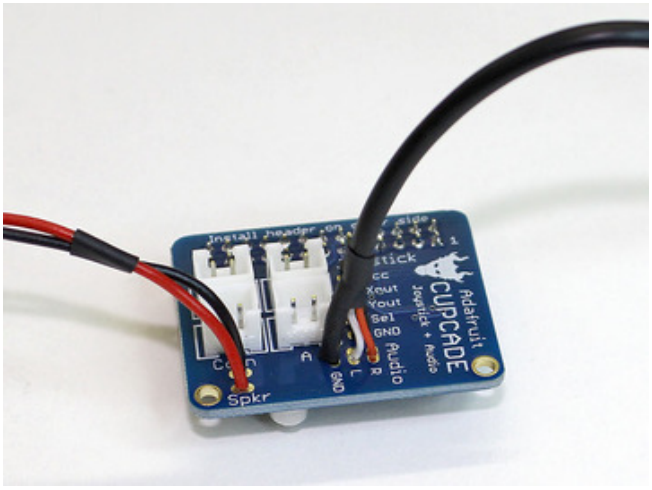
The four 2-pin JST sockets, plus the 5-pin joystick header, are inserted on the "CUPCADE" side of the board and are soldered on the back side.

It's easiest to do the JST sockets first, one at a time. You can temporarily hold them with a little tape, or place each one "legs up" on the table and lower the board in place, then solder.

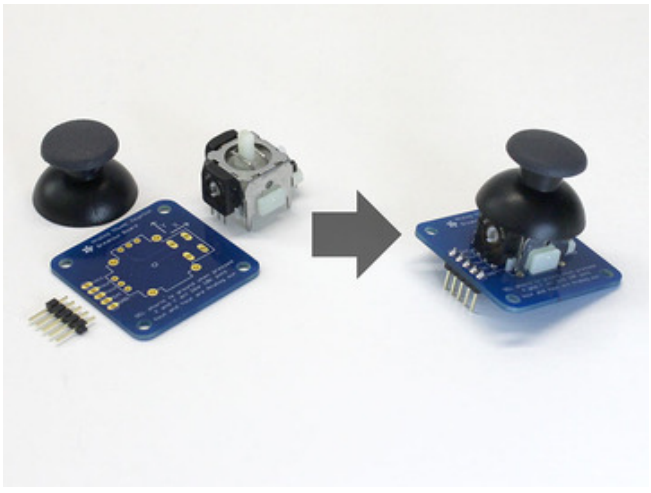
Notice how the two rows sit "back to back." This makes it a little easier to unplug the connectors later.

The 26-pin Raspberry Pi header then mounts on the underside of the board, with the pins soldered on top. **Make sure the notch on this header matches the outline on the board!**

The speaker and headphone wires then connect to

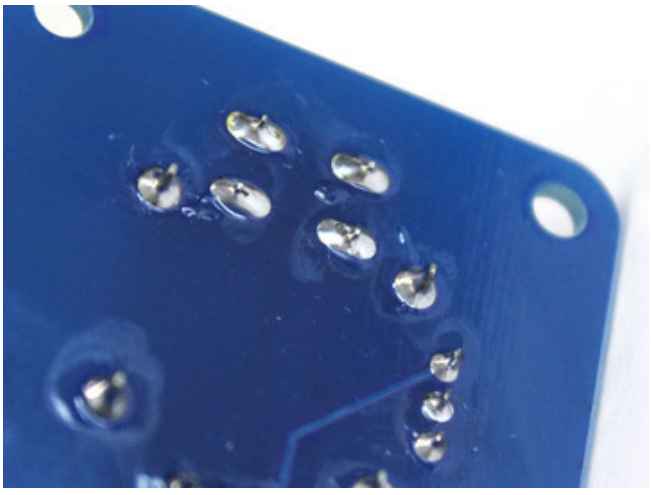


corresponding points on the board. The speaker wires go to the two “Spkr” points, headphone to the three “Audio” points: red wire to “R,” black or white wire to “L,” copper wire to “GND.”



Assemble the analog thumb joystick. Some of the pins may be a little bent from shipping and need to be nudged into place. The “hat” simply presses into place.

Install a 5-pin header from below.



Make sure all your solder connections are smooth and clean, like tiny Hershey’s Kisses®. There should be no solder balls or bridges, no gaps between pins and their corresponding pads.

Switch off your iron, all the soldering's done now!

Installing ROMs

The Cupcade does not come with games installed. However its really easy to install any kind of Arcade ROM by dragging it onto the SD card that comes with the kit. We'll show here how to install some of the free, non-commercial ROMs available from MAME.org

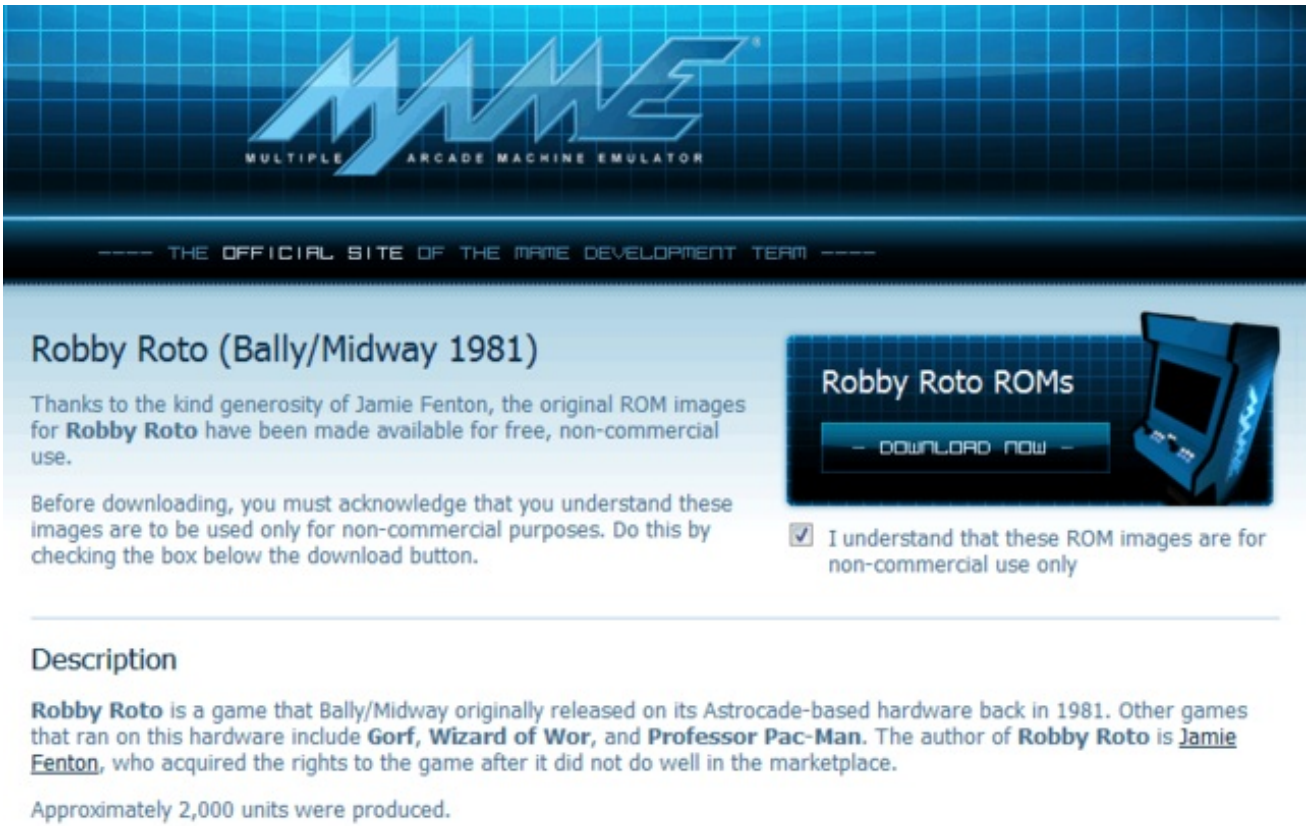
If you have other ROMs you'd like to play, just make sure they are "MAME roms" and not "NES roms" or "Sega roms". We suggest Googling for your favorite game and "mame rom"

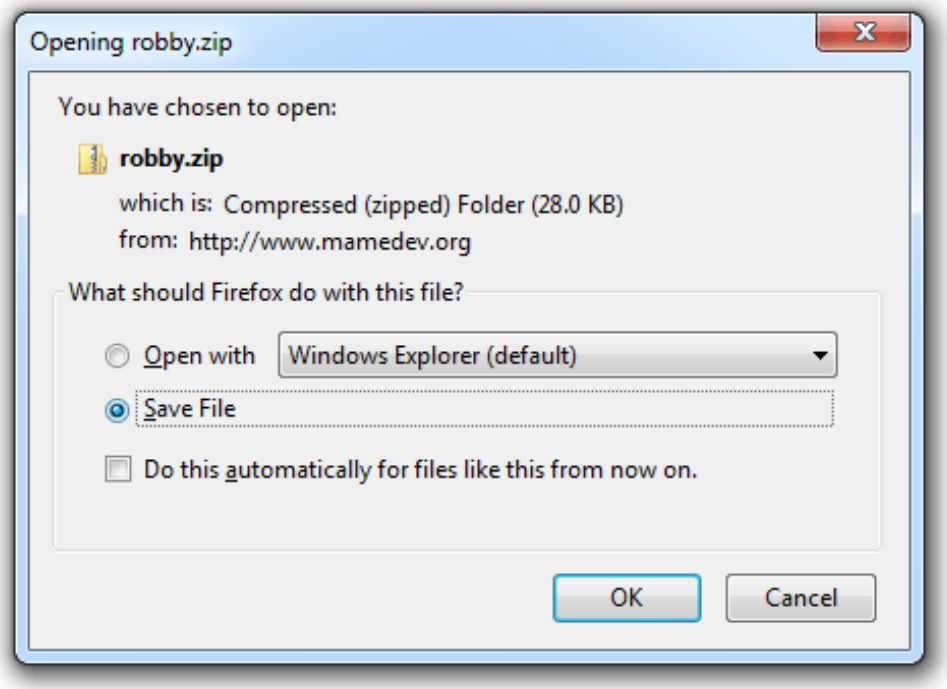
If you're feeling like an archivist, archive.org has a 42 GIGABYTE collection of every MAME ROM they could find! (http://adafru.it/dfR)

Downloading the Free ROMs

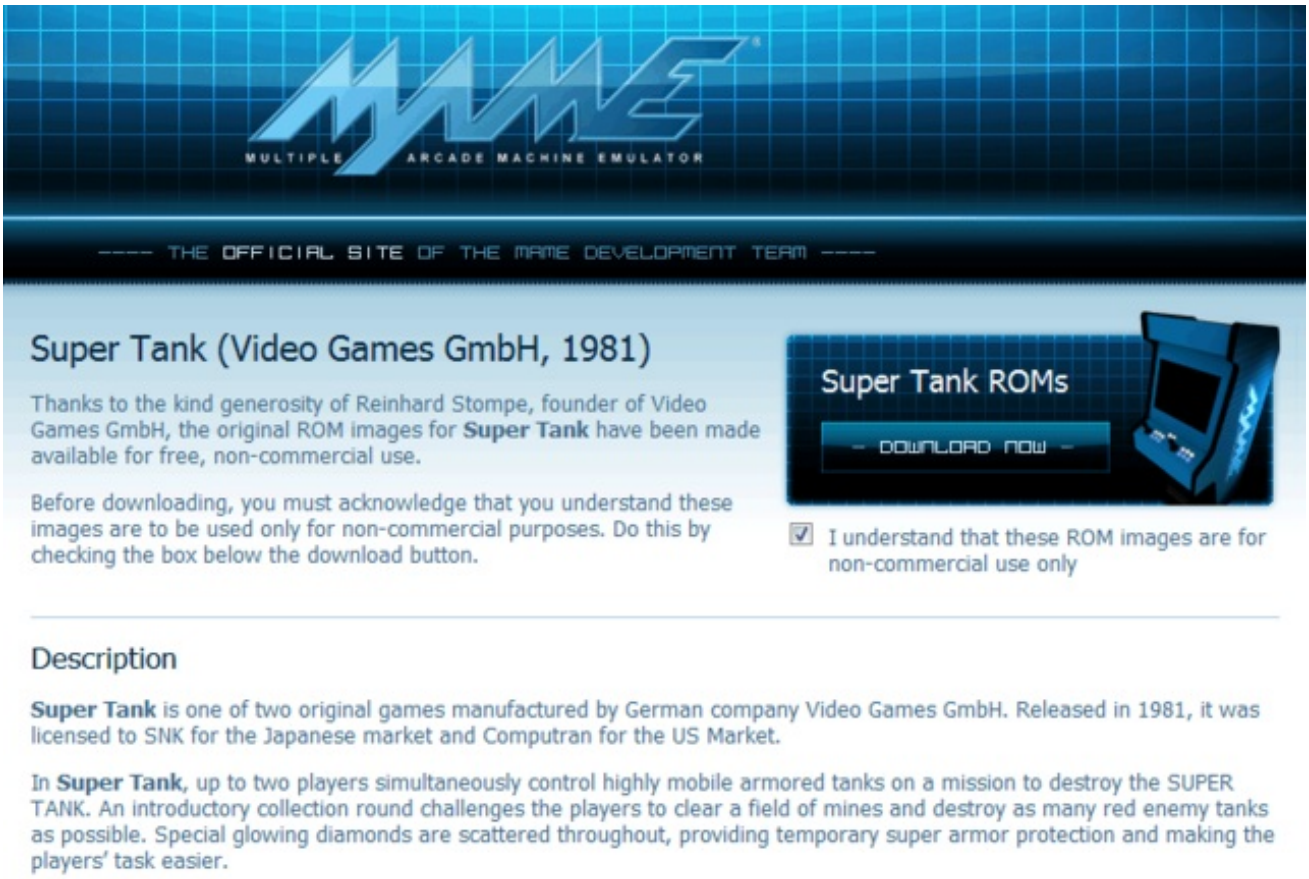
Before proceeding with your dry run, you'll want to have a game to play! There are a few free non-commercial ROMs available for testing your setup available. Visit the page at http://mamedev.org/roms/ (http://adafru.it/dfS) and download the **Robby Roto** (horizontal video) or **Super Tank** (vertical video) ROMs.

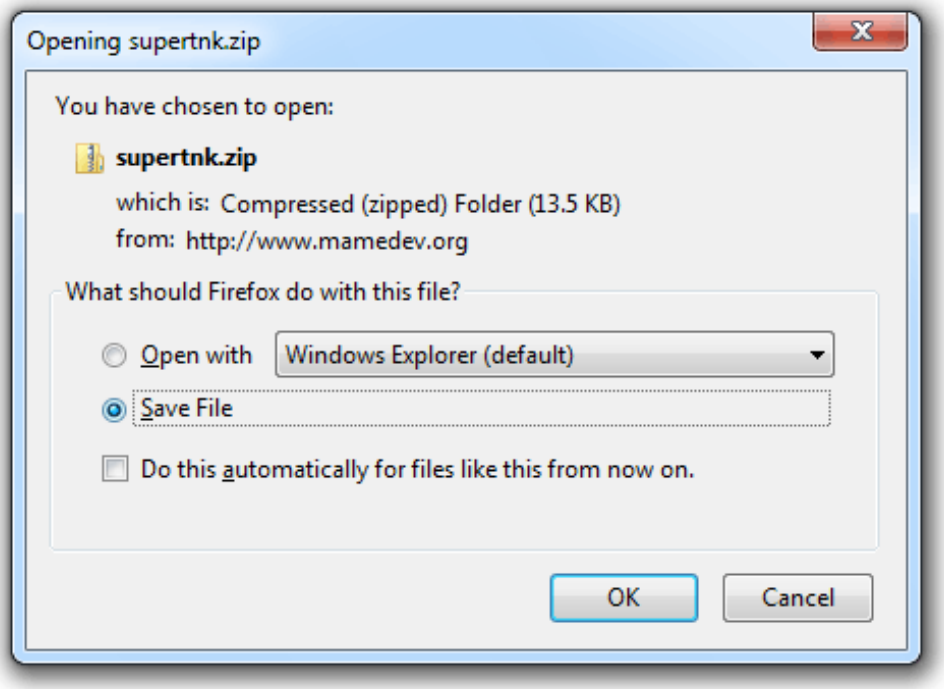
Neither runs quite perfectly on the Cupcade (they use different screen resolutions) but are sufficient for testing the sound, buttons and joystick.


MULTIPLE ARCADE MACHINE EMULATOR
THE OFFICIAL SITE OF THE MAME DEVELOPMENT TEAM
Robby Roto (Bally/Midway 1981)
Thanks to the kind generosity of Jamie Fenton, the original ROM images for Robby Roto have been made available for free, non-commercial use.
Before downloading, you must acknowledge that you understand these images are to be used only for non-commercial purposes. Do this by checking the box below the download button.
Robby Roto ROMs
DOWNLOAD NOW
I understand that these ROM images are for non-commercial use only
Description
Robby Roto is a game that Bally/Midway originally released on its Astrocade-based hardware back in 1981. Other games that ran on this hardware include Gorf, Wizard of Wor, and Professor Pac-Man. The author of Robby Roto is Jamie Fenton, who acquired the rights to the game after it did not do well in the marketplace.
Approximately 2,000 units were produced.



Don't decompress the ZIP file! Keep the ROM files in the ZIP!

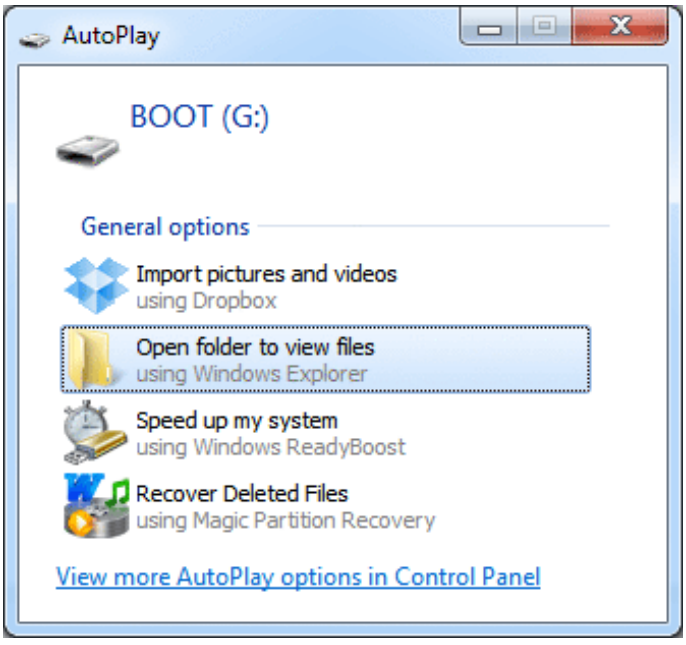

MULTIPLE ARCADE MACHINE EMULATOR
THE OFFICIAL SITE OF THE MAME DEVELOPMENT TEAM
Super Tank (Video Games GmbH, 1981)
Thanks to the kind generosity of Reinhard Stompe, founder of Video Games GmbH, the original ROM images for Super Tank have been made available for free, non-commercial use.
Before downloading, you must acknowledge that you understand these images are to be used only for non-commercial purposes. Do this by checking the box below the download button.
Super Tank ROMs
DOWNLOAD NOW
I understand that these ROM images are for non-commercial use only
Description
Super Tank is one of two original games manufactured by German company Video Games GmbH. Released in 1981, it was licensed to SNK for the Japanese market and Computran for the US Market.
In Super Tank, up to two players simultaneously control highly mobile armored tanks on a mission to destroy the SUPER TANK. An introductory collection round challenges the players to clear a field of mines and destroy as many red enemy tanks as possible. Special glowing diamonds are scattered throughout, providing temporary super armor protection and making the players' task easier.



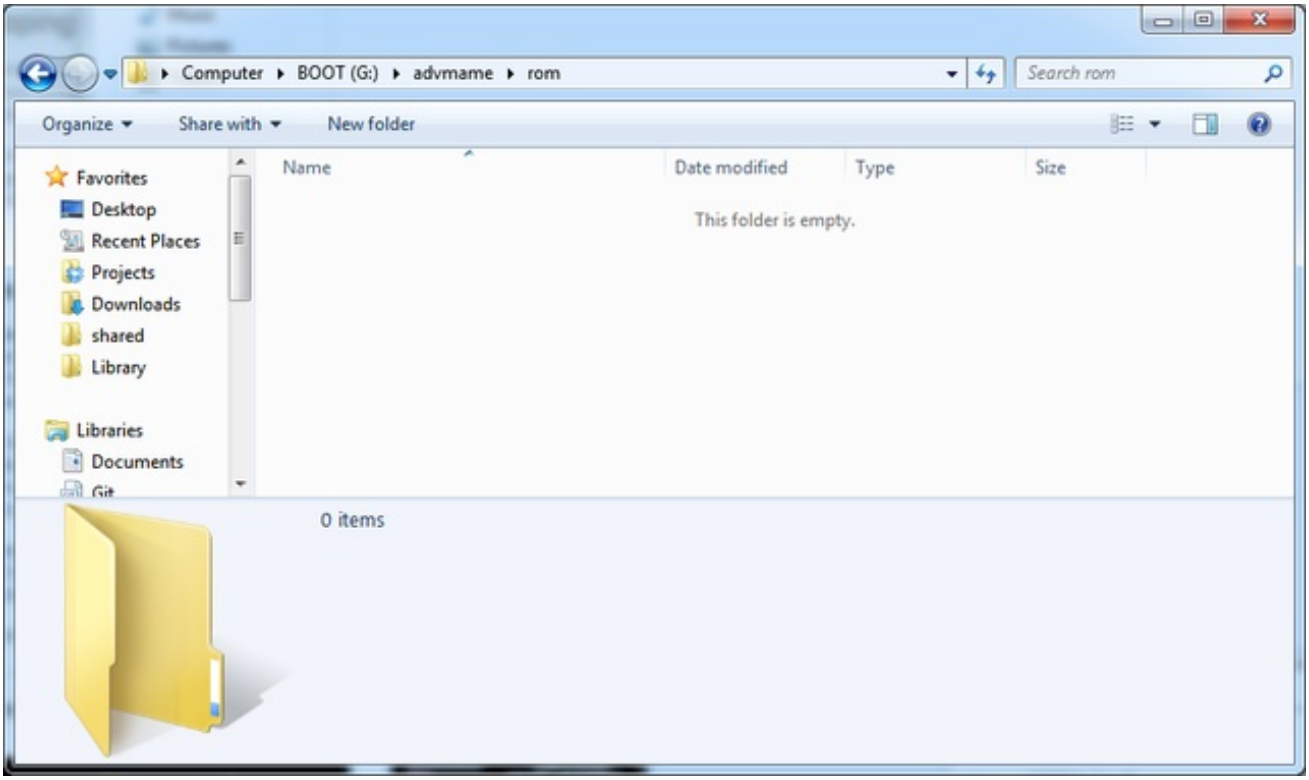
Don't decompress the ZIP file! Keep the ROM files in the ZIP!

Installing ROMs

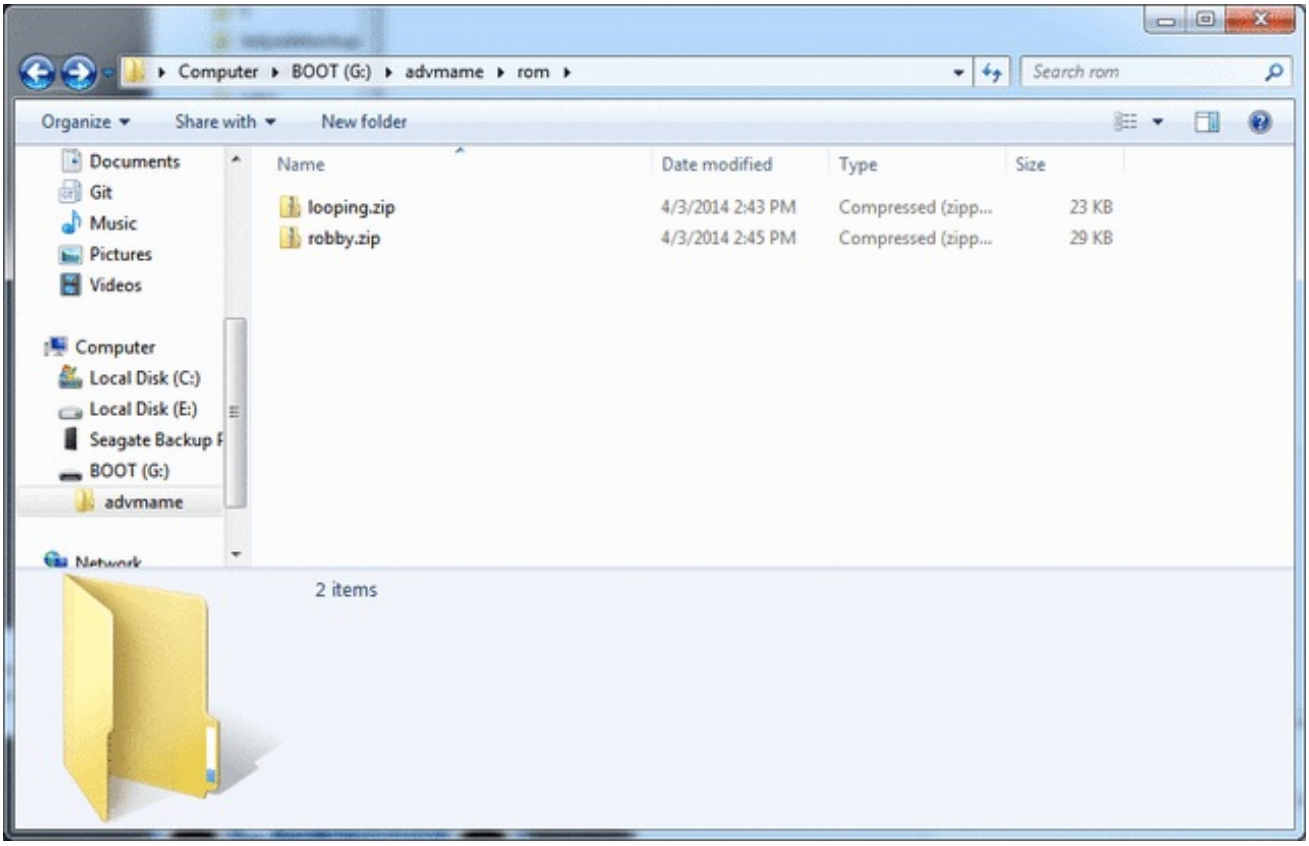
Now you have them, you can install the ROMs. Insert the cupcade SD card into any computer and browse the contents of the BOOT partition



Navigate to the **advmame/rom** folder, it should be emprty



Drag the **zip** files of the ROMs into the **roms** folder as seen here:



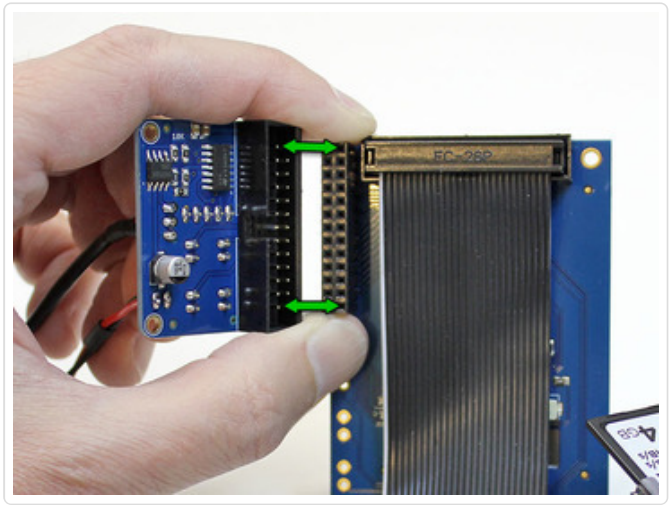
Safely eject the SD card! Now you can plug it into the Pi for your dry run.

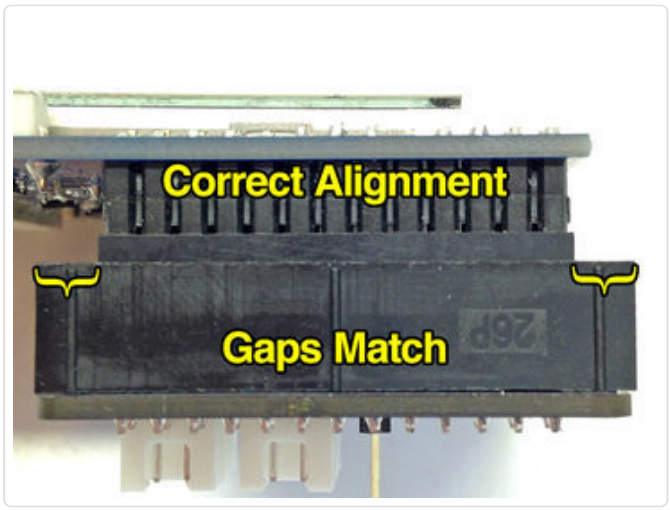
Dry Run

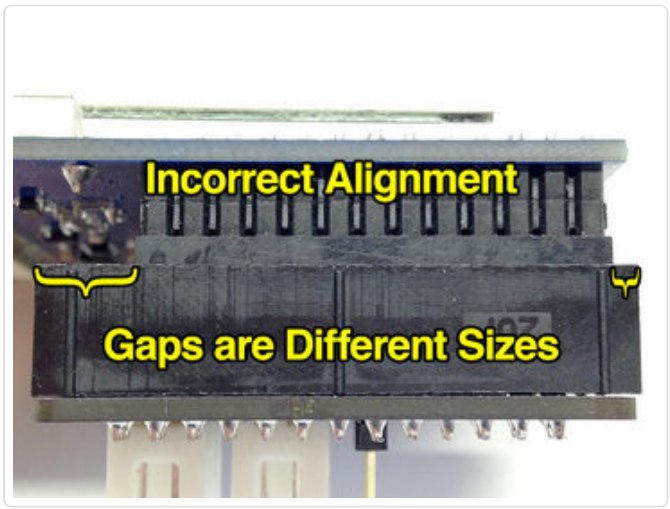
Re-install the ribbon cable between the Raspberry Pi and PiTFT, making sure the Pin 1 (white wire) is in the correct location.

The Cupcade interface board now installs on the female header on the back of the PiTFT.

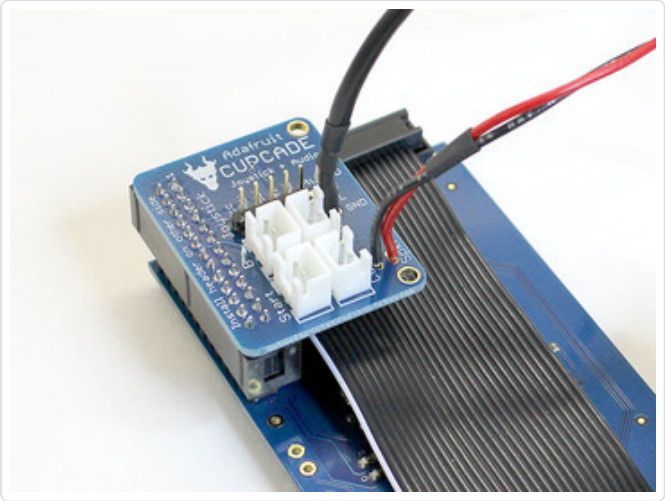
Be super careful to line up the pins! There's enough wiggle room either way for this to be offset by one pin. The Pi won't boot unless it's properly centered.

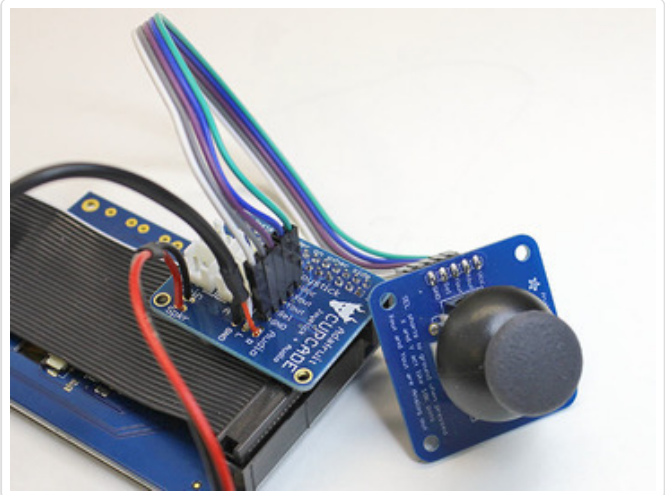






The board should be oriented to sit behind the PiTFT and ribbon cable, not hanging out over the side.





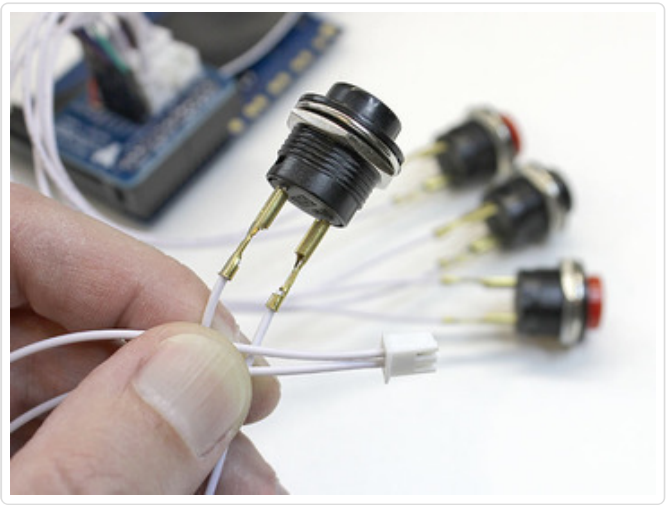
Connect the female jumper wires between the analog thumb joystick and the Cupcade board, making sure the same pins are connected at each end (e.g. GND to GND and so forth). Wire colors vary from batch to batch; yours might not match this photo, that's OK.

The Sel pin is not used. If your kit has only four wires in this bundle, just skip over that pin.

The joystick MUST be connected for the Cupcade to work.

Plug a quick-connect wire set into each of the four buttons.

There's no specific polarity to these…either wire can go to either pin.



The two red buttons will serve as the “A” and “B” action buttons. The two black buttons are for “insert coin” and “1 player start” (we’ll refer to them as **¢** and **1P** for short).

Plug each quick-connect into the corresponding labeled jack on the Cupcade interface board.

Plug in the **audio cable** and you’re almost ready do go…

Before connecting power, make sure no conductive parts are making accidental contact with each other:

- Quick connects or the nut on each button
- The metal frame of the speaker
- The back side of the Cupcade interface board
- USB and Ethernet jacks on the Raspberry Pi

You can tape pieces down to your desk if it helps.

Once everything's safely spaced apart, plug in the Pi. It should boot to the game menu.

If the Pi Does Not Boot

If you don't see the system starting the boot process within 10 seconds or so (watch the ACT LED on the board), disconnect power immediately.

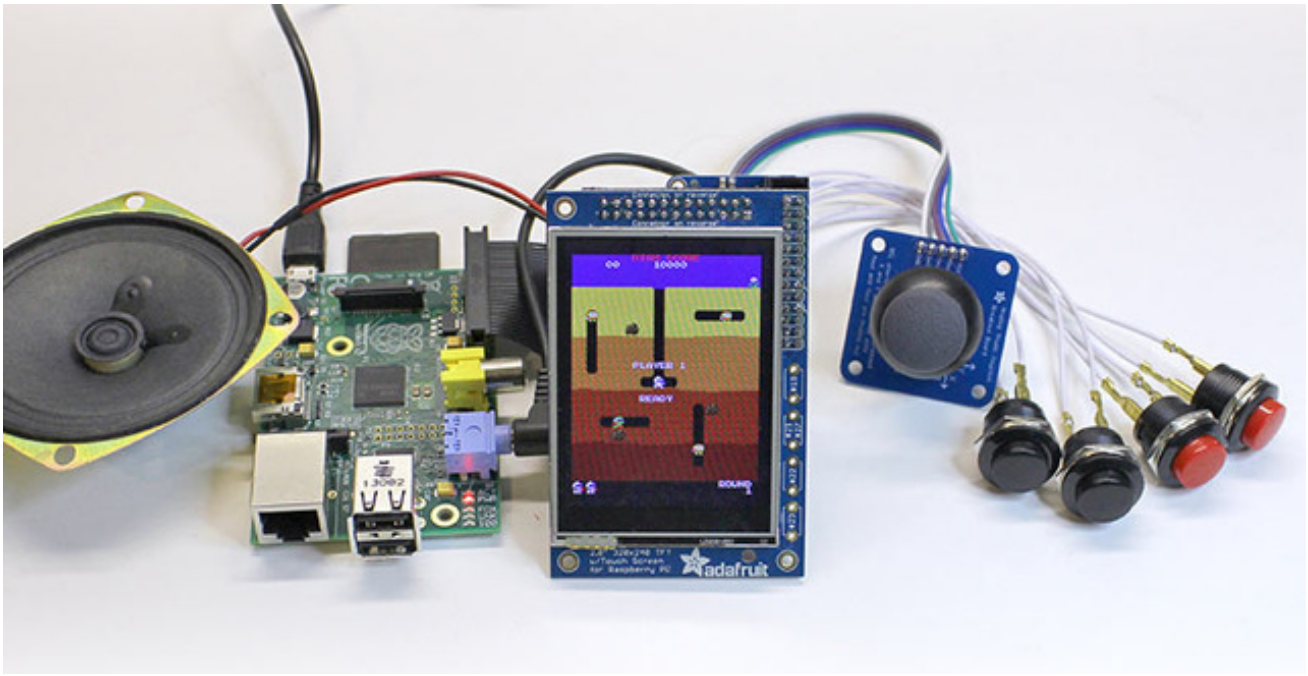
Carefully unplug the Cupcade interface board from behind the PiTFT and try booting again. Does it work now? If so, the problem is usually with the new parts.

- Was the board properly aligned with the header on the PiTFT? It's easy to get this off by one.
- Are any of the button quick-connects making accidental contact with each other?
- Is the joystick connected? Cupcade won't boot without it.
- Are the four wires connected to the proper pins in the correct order on the joystick? Even a single mislaid wire here can prevent the Pi from booting!
- Examine the joystick closely, specifically the horizontal and vertical potentiometer dials on the sides. When you move the stick around, can you see both dials turning with it? If not, the joystick may be faulty.

If the Pi still refuses to boot when the board is connected, visit the Adafruit Customer Support Forums (http://adafru.it/cer) for help. Post clear photos showing your wiring and soldering and we'll look it over for trouble spots and make recommendations.

If the Pi Boots Successfully

Rejoice! Test the game menu by moving the joystick around. If you have any ROM files loaded in the /boot/advmame/rom folder, you can select one and try playing with the controls instead of a keyboard.



When You’re Done Testing

If you have a game loaded, hold down both the **¢** and **1P** buttons for a few seconds. This will either return to the game menu, or will pop up a Continue/Exit menu, which can be navigated with the joystick and **A** button.

From the game selection menu, hold down both the **¢** and **1P** buttons for a few seconds to initiate an orderly shutdown. Wait for the “halted” message before unplugging.

Or, if you have further system configuration to do, connect a USB keyboard and press **ALT+F3** for a login prompt.

Do not continue until the system is tested and the controls are known working.

Cabinet Part 1

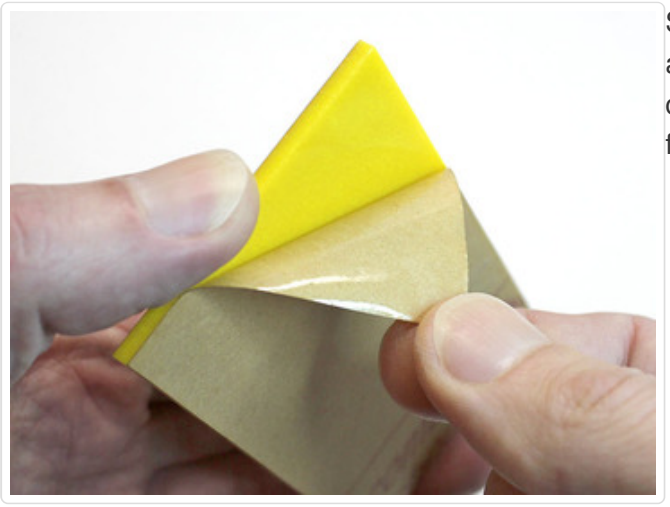
Fitting all the electronics inside the case is the most challenging part of this project. Our advice:

- **Don't force anything.** If something refuses to fit, it might simply be in the wrong position.
- You might need to **temporarily disconnect** some parts (like the buttons or joystick) to re-route the wires in a less tangled manner. Sometimes repeatedly. If the wiring seems to magically change from one of our photos to the next, that's exactly what happened.
- **Take your time.** If things aren't cooperating and you get frustrated, walk away, try again after a break.

If this is such a pain, why didn't you just design an easier case to build?

It makes sense once it's all together. There are just a few visible screws on the sides, and the area around the controls is perfectly smooth, with no screw heads scratching at your fingers. An easier case would have a lot more visible fasteners, some of them in uncomfortable positions!

Like the electronics, the case assembly starts with prep work. With all our ducks in a row, the challenging part becomes slightly less challenging.

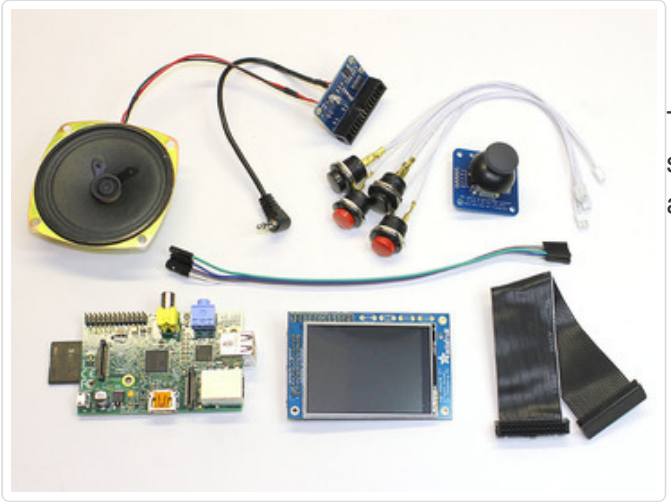


Start by peeling the backing paper off both sides of all the laser-cut parts. It's easiest to start at a corner, catching the edge of the paper with a fingernail.

The laser-cutting process sometimes leaves a little paper soot at the edges. If you like, you can wash these off with soap and water, just be **absolutely certain** that all the parts are **completely dry** before proceeding!

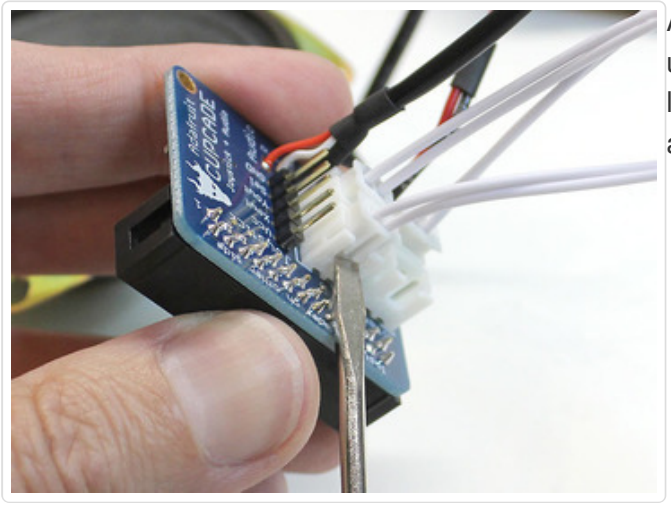
With the Pi powered off, disconnect all of the separable parts:

- Raspberry Pi board
- PiTFT display
- Ribbon cable



- Joystick
- Buttons (4)

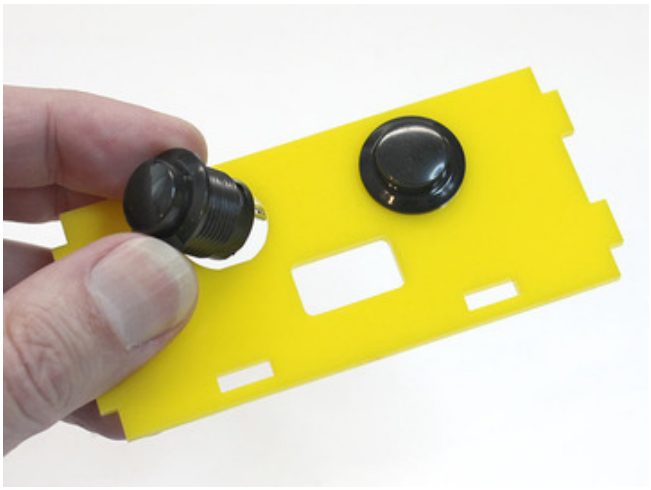
The Cupcade board, speaker and audio cable should all stay joined as a single thing; don't clip any wires!



A small screwdriver is sometimes helpful when unplugging the buttons from the JST sockets. A little twist is usually sufficient…use common sense and be gentle.

Remove all of the nuts and washers from the four buttons. You can leave the quick-connect wires attached.

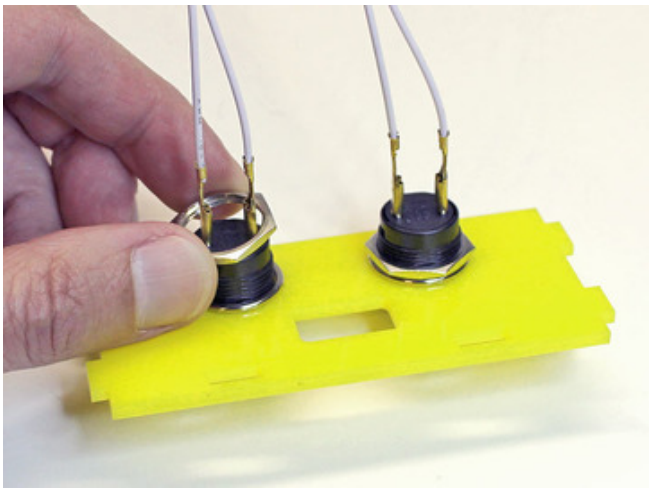
Lighting and staging photos is time-consuming! A handful of images from beta units are still shown here. Don't panic if your Cupcade looks a little different…it should all work the same.

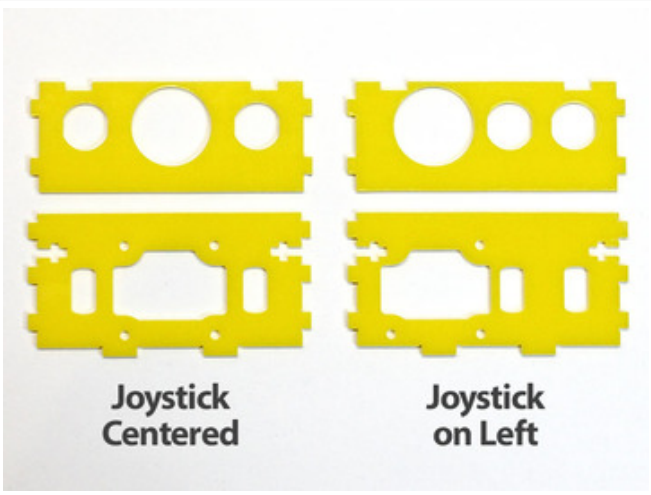


Let's do the credit and start buttons first. We chose the black buttons for this.

Look for the acrylic piece that resembles a startled face. Slide a button (with attached quick-connect) through each "eye," then add a washer and nut from behind.

Tighten these with finger pressure only. Don't use tools or you'll crack the plastic!

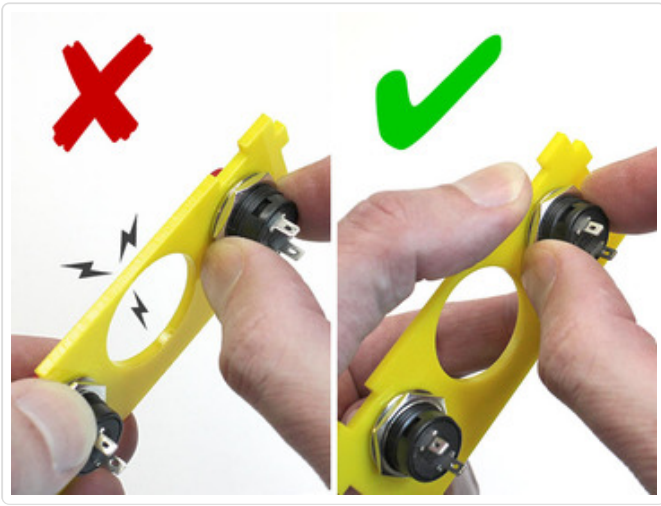




Early on, we mentioned a choice to be made: centered or off-center joystick? Now is your last chance to decide.

There are two acrylic parts required for the controls. Select the pair that matches your desired control layout. The other two can be stored somewhere in case you want to switch it out later.

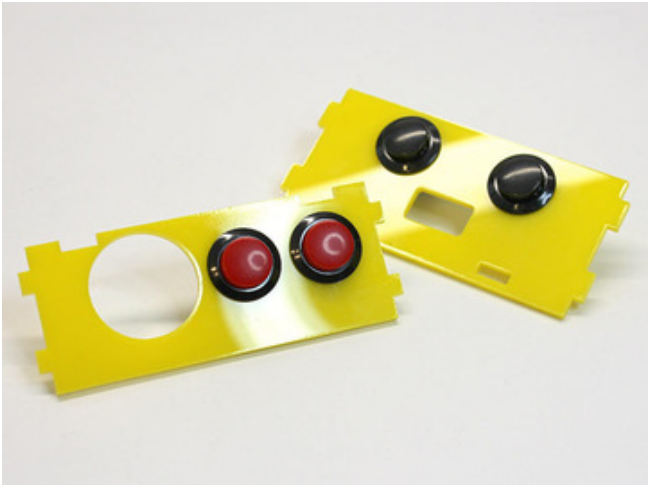
If you want the joystick on the right, just flip these pieces over.



Two red buttons are installed on the chosen control cutout. Because this part is thin in one spot, it must be done with care.

When tightening each nut, hold the acrylic piece close to the corresponding button, not at the far end.

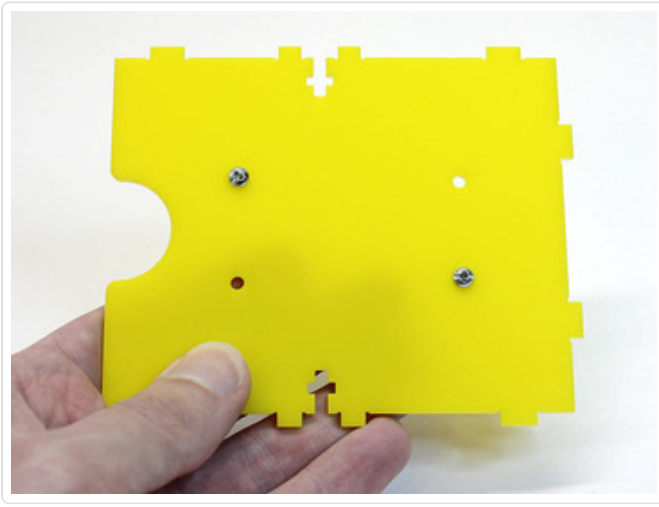
(This is one of those beta photos. Yours will have the quick-connects attached.)



Taa-daah!

Notice the piece with the red buttons has two "bites" along the **top edge**. If they're at the bottom, and if you're using an off-center joystick layout, undo the buttons and flip this piece over. With a centered layout, just turn it around.

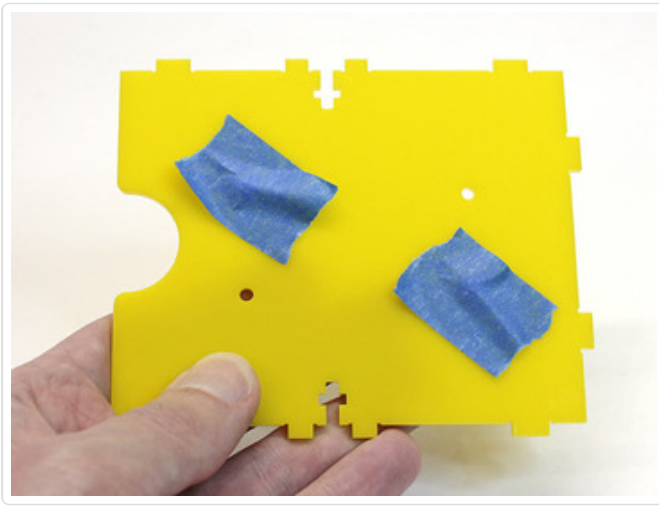
Masking tape. You'll need it. Doesn't matter if it's the blue or tan type, but it does need to be masking tape. It has a relatively weak grip and doesn't leave residue behind.



Locate the acrylic bottom piece and insert two #4-40 1/2" screws in the positions shown.

There are four holes, but only two screws. These align with mounting holes on the Raspberry Pi board. Most of the case pieces are symmetrical so they can be flipped either way…makes things a little easier.

Add a piece of masking tape over each screw head to keep them from falling out.



Set the piece down on your work surface with the screws pointing **up**, then add a nylon spacer over each screw.

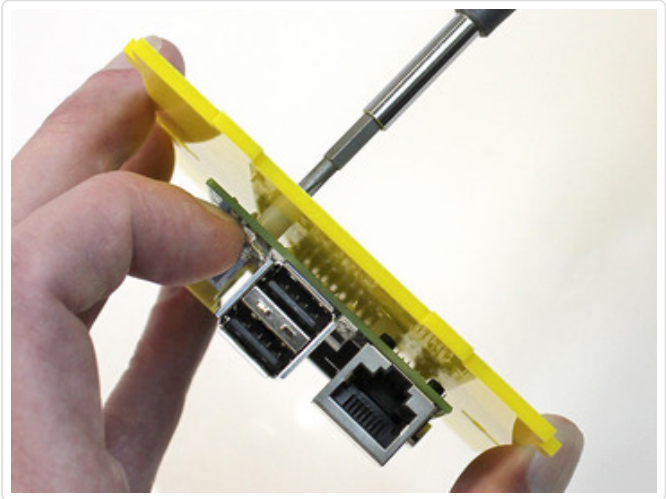
Align the mounting holes on the Raspberry Pi over

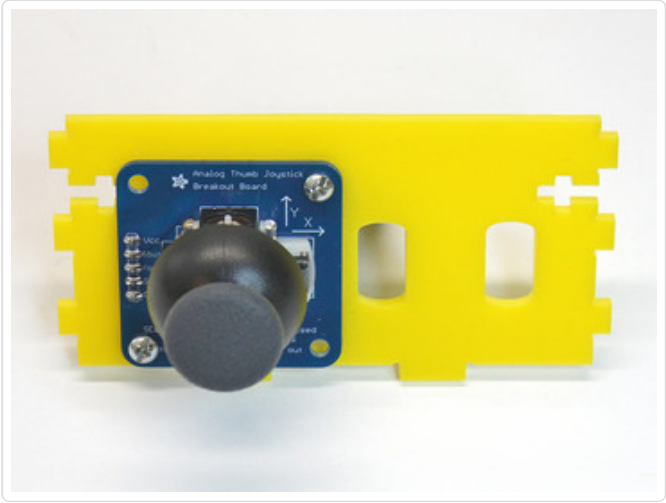
these two screws.

The SD card slot should be facing the edge with the "bite." If you get this backwards, there will be pain and anguish later as everything has to be dismantled and turned around.

Add a nut on each screw and give them just a few easy turns.

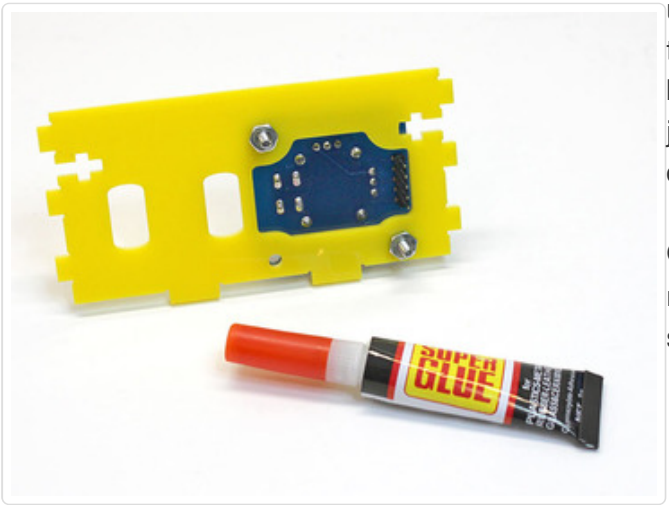
Once the board is held in place, remove the tape and give each screw an extra half turn or so with a screwdriver. Don't go overboard, you don't want to crack the plastic, just make sure the board is secure and the screws aren't rattling.





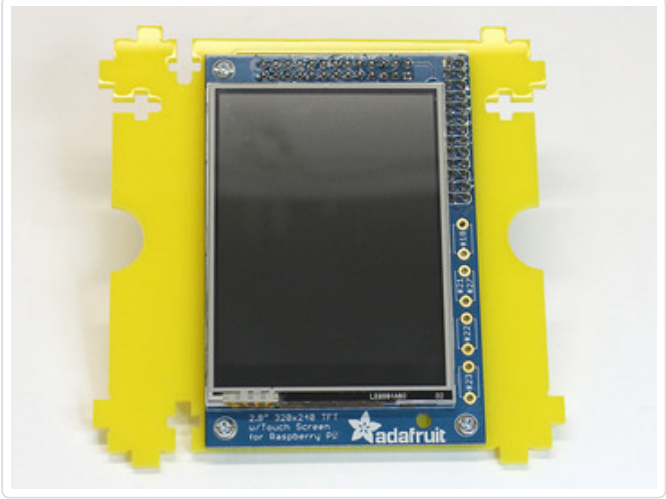
Install the joystick on its support piece, whichever one you selected (centered or off-center). Two screws at opposite corners are sufficient.

Note the orientation of the joystick and support piece here. With the joystick's silkscreen labels upright for reading, the support's two protruding tabs should be at the **bottom** of the piece. This is one of the few non-symmetrical parts, so make sure you get the orientation right. It's very frustrating to do over.



There's a little wiggle room between the parts. If using an off-center joystick (as shown here), scoot the joystick that tiny extra bit *away* from the buttons before tightening the screws. This keeps the joystick from bumping up against the buttons during play.

Optional: a dab of thread lock, super glue or hot-melt adhesive on each nut helps keep these screws from loosening with heavy gameplay.

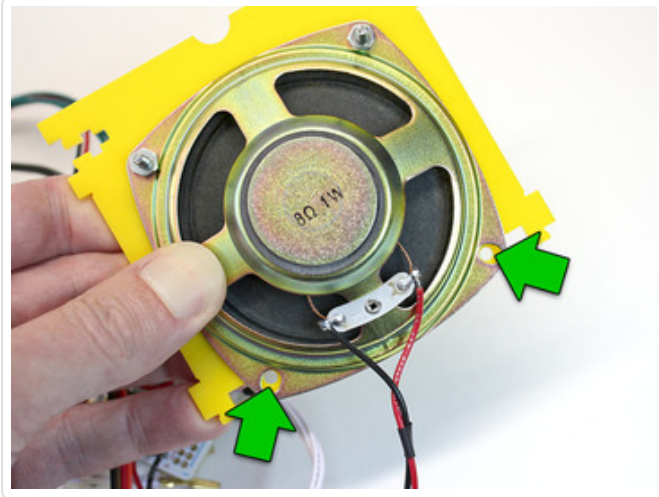


Install the PiTFT on its backing piece using three tiny #2-56 screws and matching nuts.

This is another non-symmetrical part with a specific orientation. The screw holes should make it evident which way this goes.

Install the speaker grille using two screws at the **top**.

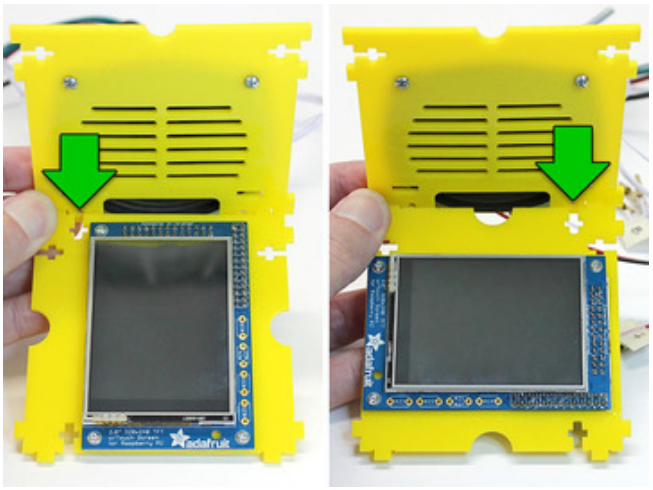
The bottom holes do not receive screws (yet). However, before tightening the top nuts, make sure the bottom screw holes in the speaker and grille are aligned; there's a bit of play.



Awesome. Now the puzzle box stage begins. This is a good time for a cookie break.

Cabinet Part 2

Now we'll join the speaker and screen parts…

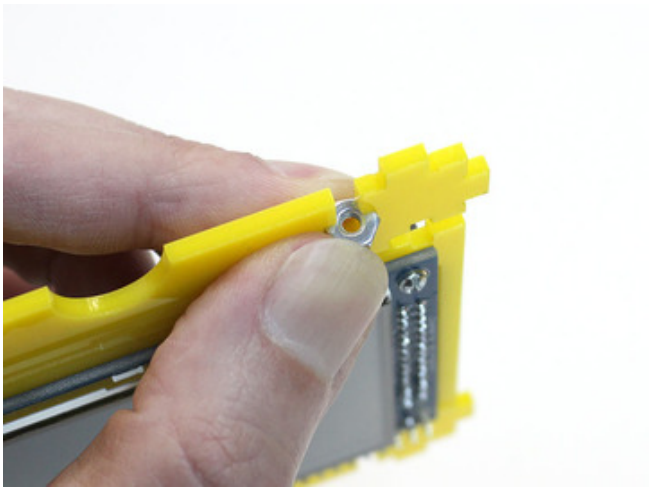


The screen fits into two slots at the bottom of the speaker grille.

Depending which way you've installed the screen — horizontal or vertical —one "T-slot" will be located at either the left or the right side.

For a **VERTICAL SCREEN:** the T-slot should be on the **LEFT.** If it's on the right, you've got the screen piece upside-down.

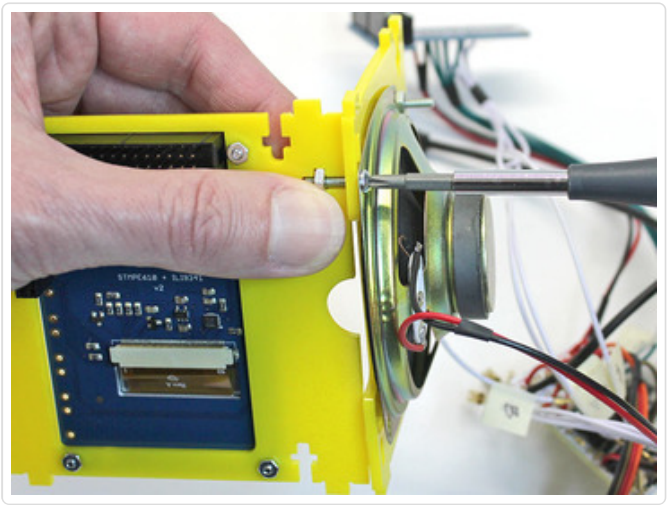
For a **HORIZONTAL SCREEN:** the T-slot should be on the **RIGHT**. If it's on the left, the screen piece is upside-down.



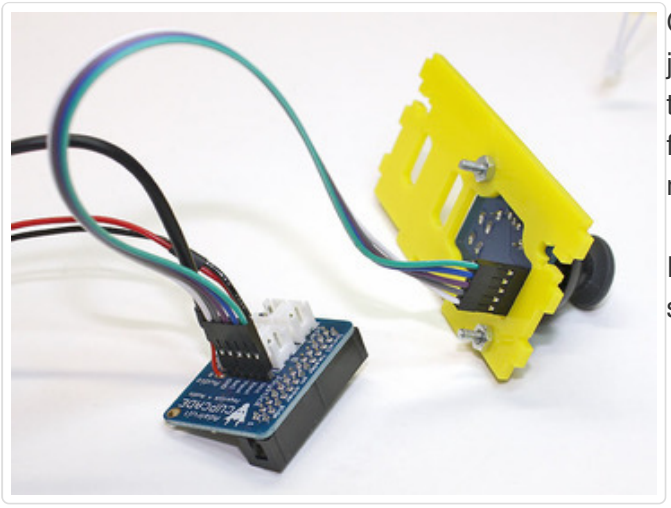
Feed a nut into the cross part of the "T," holding it in place between thumb and forefinger.

Bring the two pieces together and add a #4-40 1/2" screw. This will pass through the speaker frame, the plastic grille piece and into the nut.

Only one of the two speaker/screw holes is used, depending which way the screen is oriented.



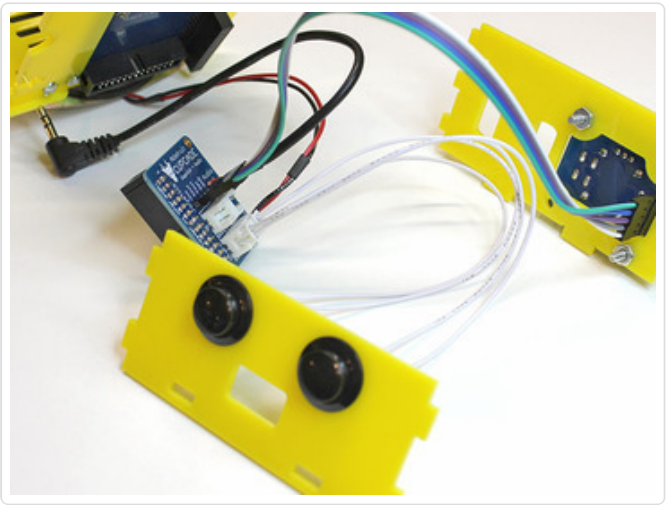
Joystick and Buttons

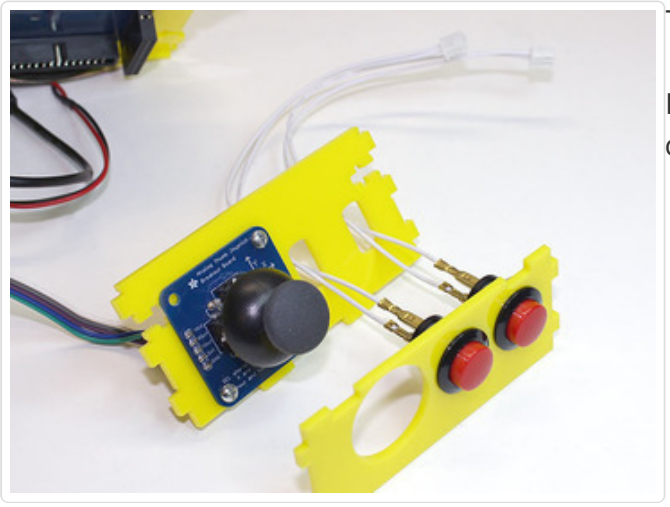


Connect the female jumper wires between the joystick and Cupcade board, making sure you get the same sequence (e.g. GND to GND and so forth). If you get these out of sequence, the Pi might not boot!

If your wire bundle has only four wires, you can skip over the **Sel** pin; it's not being used here.

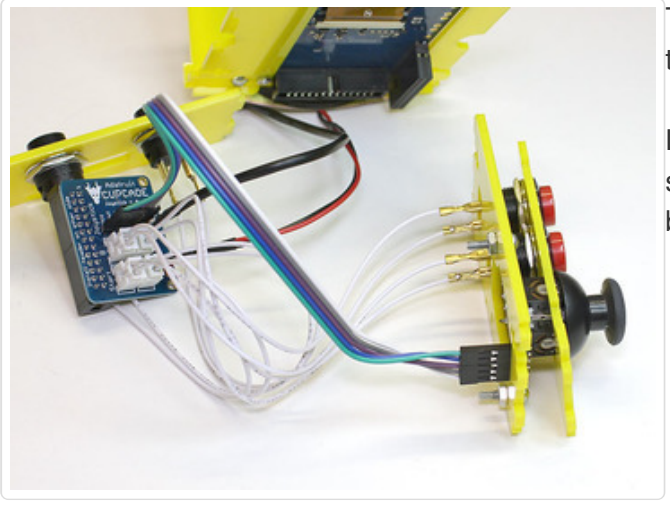
Plug the **¢** (left) and **1P** (right) buttons into the corresponding sockets on the Cupcade board.





The **A** and **B** buttons require an extra step…

First, feed the quick-connect wires through the corresponding slots on the joystick plate.

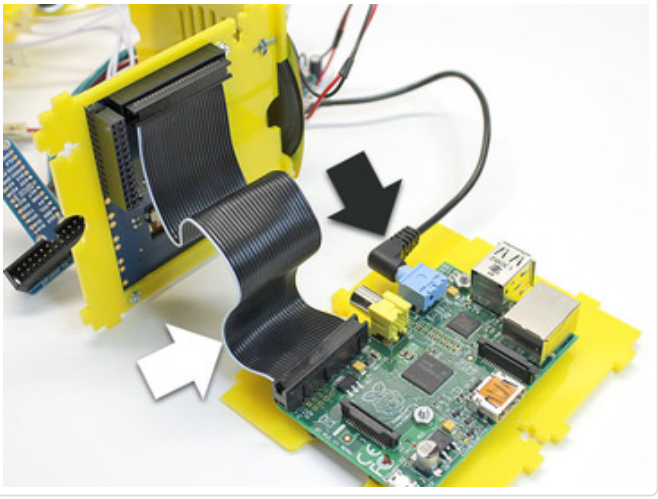


Then plug these into the corresponding sockets on the Cupcade board.

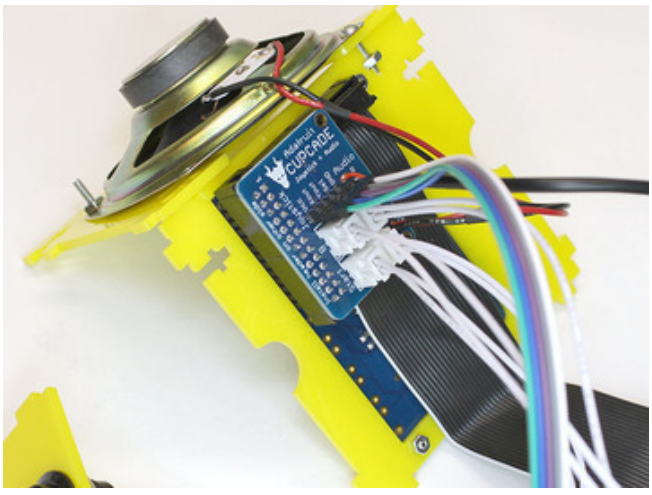
Loosely fit the button support over the joystick support. It's okay if this flops around for the time being…everything will be held in place later.

Look over your wiring before proceeding. Are the wires reasonably well organized, or are they twisted around each other like weeds? If necessary, unplug one wire at a time, untangle it from its neighbors and plug it back into the correct location. This isn't just a persnickety thing, it's actually important for fitting everything in the case later!

Screen and Audio



Plug the audio cable into the headphone jack and the ribbon cable to the GPIO header, making sure pin 1 (the white wire) is in the correct place.



The Cupcade board plugs into the back of the PiTFT as it was during our dry run.

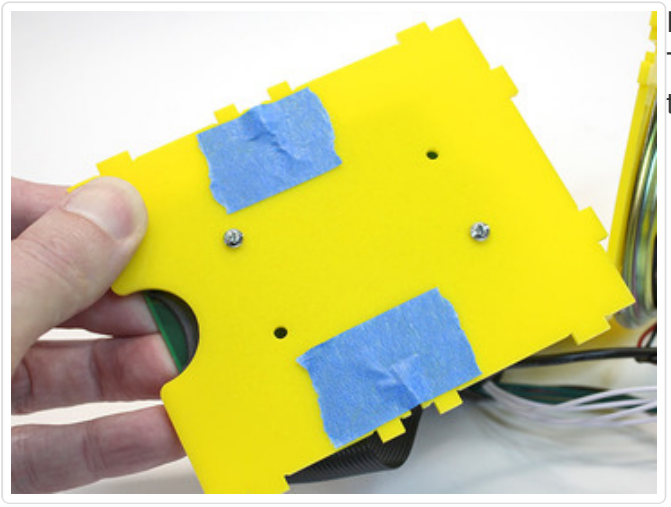
Be super extra careful to **get the headers correctly aligned.** There's enough wiggle room either direction for it to go one pin out of alignment!

You can do another dry run at this point if you like…it's a really good idea. Then shutdown the system (hold credit + start buttons while the game menu is showing).

Cabinet Part 3

The case uses the “T-slot” assembly technique, with #4-40 screws and nuts. You saw one of these when joining the screen and speaker pieces.

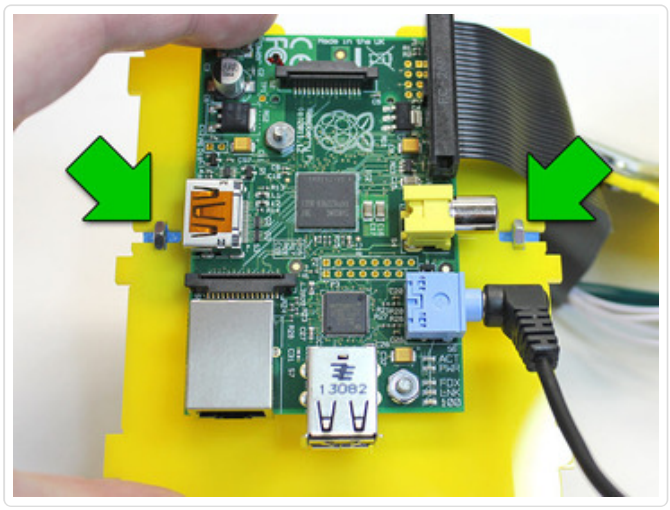
Unlike that first slot, the remaining T-slots will be too deeply recessed to reach with fingers. Instead, we’ll **use masking tape to temporarily hold each nut in place**, then remove it later.

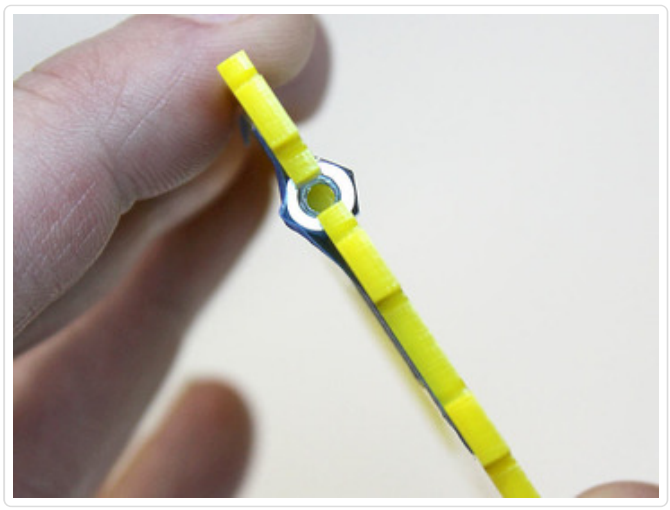


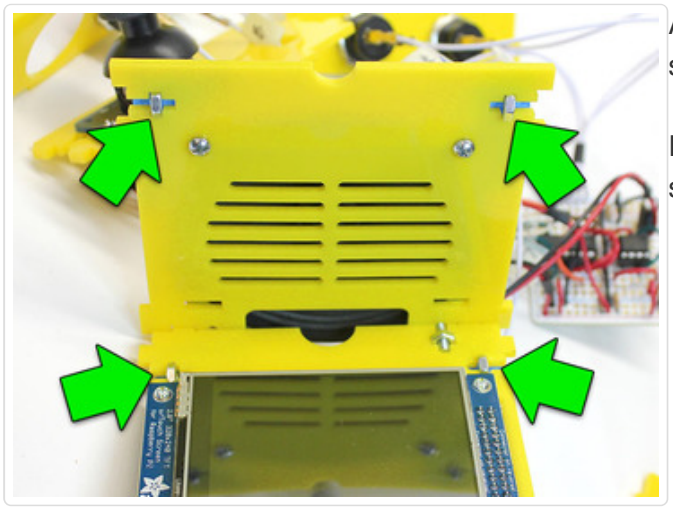
Put a piece of masking tape over each of the two T-slots on the underside of the base piece (where the Raspberry Pi is installed).

Then, from the top side, press a nut into the cross part of each “T.” Give it a pinch from both sides, so the tape gets a good hold on the nut.

Try to get the nuts roughly centered-*ish* in the slots.



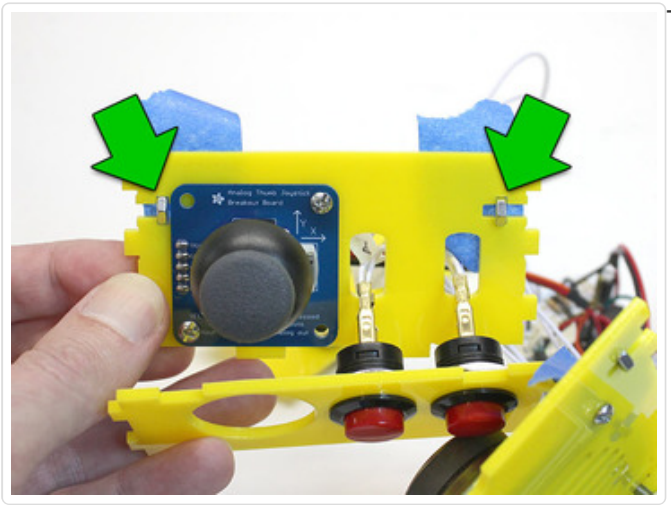




Add tape behind the two T-slots at the top of the speaker grille, then press nuts into place.

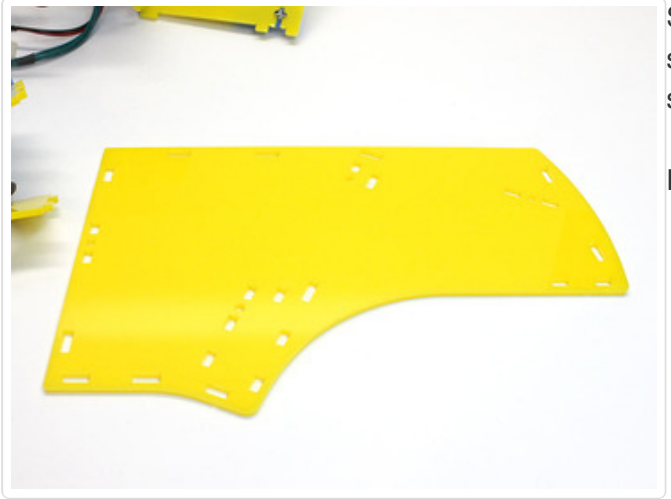
Do the same for the two slots near the top of the screen.

And once more for the joystick support piece.



There should be 8 nuts & tape bits at this point.

Okay, the Tricky Part



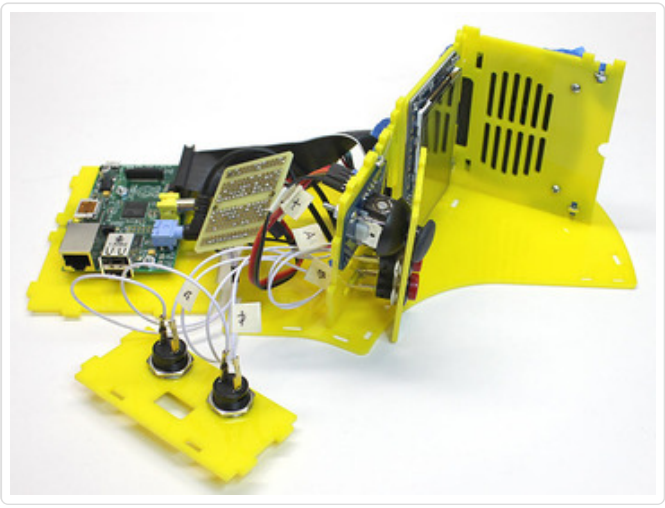
Set one of the large side pieces flat on your work surface. Either one is fine…we're using the right side here.

Hey! That wasn't so hard. Wait for it…

Take the jumble of parts and try to (roughly) align the tabs on the speaker grille, screen and joystick supports with the corresponding holes in the side piece. They probably won't *stay* in those holes for very long, but at least you'll know where they belong.

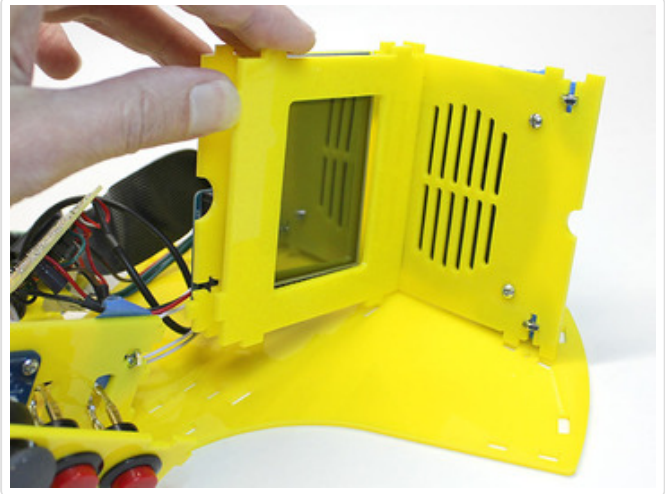
Start to place other elements vaguely in position: Pi near the bottom, buttons near the front.

If you find your wires getting tangly again, it's okay



to disconnect and re-route them.

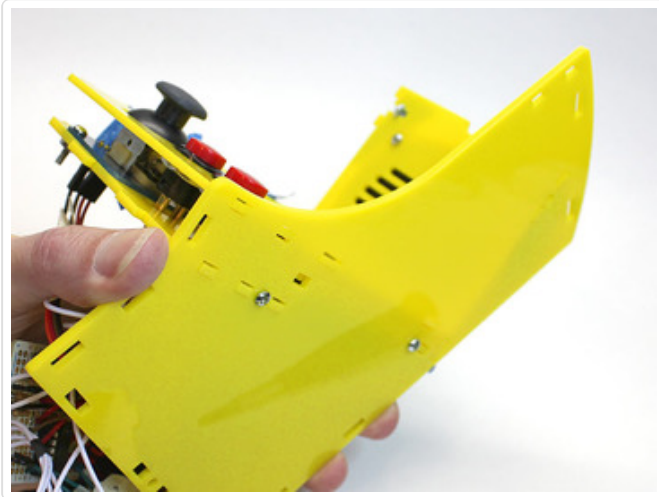
And so begins the interpretive dance called *Sealing Up the Cupcade Case…*



Pivot the screen support piece up slightly, fit the screen bezel into the slots on the speaker grille, and lower it all back into place.

Fit the joystick and button supports into their own slots as you're working on this section. **Use masking tape to hold these four pieces in alignment.**

Notches along the top edge of the button support should align with tabs on the screen bezel. If not, your button piece was assembled upside-down. Remove the quick-connects, unscrew the buttons, turn the piece over and reassemble.



The trick now is to pick the whole thing up and feed these two screws through their corresponding holes in the side.

The screws don't need to be tight…in fact a little "give" is helpful at this stage. They just need to catch the nuts that are taped in place.

Some of the acrylic parts will fight a bit and not fit into their slots. Poke a small screwdriver through the hole and try to nudge these parts the right way.

If this is a dexterity challenge or you have smaller hands: rather than installing screws from below, remove the masking tape from the prior step and try fitting the side into place from above. Once the screws catch, you can turn it over and proceed through the same steps (just reversing right and left). Another approach is to scoot the whole assemblage off the edge of the table just enough to reach the screw holes from below (one at a time).

A **third screw** installs from the side at the top of the speaker grille.

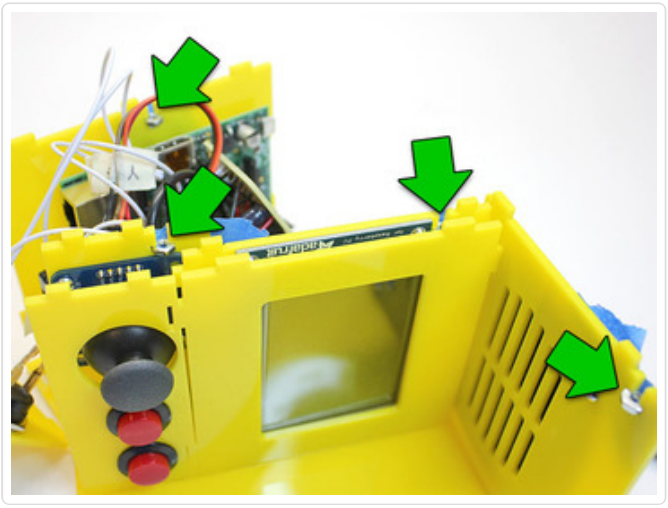
A **fourth screw** holds the Raspberry Pi base. You'll need to stand this base piece up and fit it into the slots at the bottom edge of the side piece.

You can now remove the tape from a couple steps prior. Keep the other tape (holding nuts) in place for now.

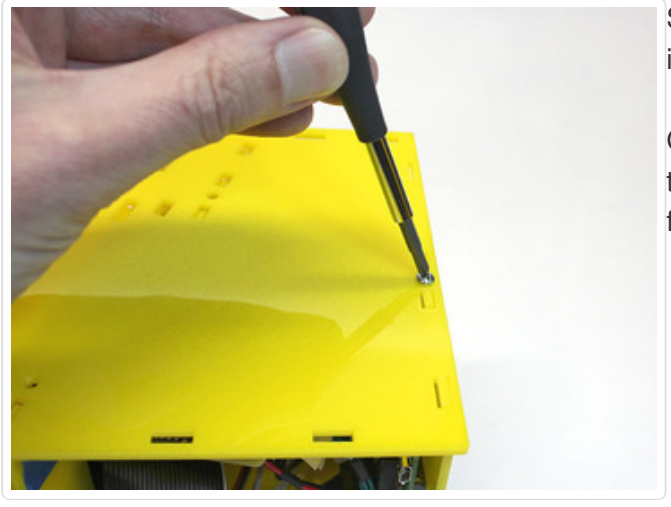
The plastic piece holding the credit and start buttons should be hanging out the front of the case at this point. If not, move it there. It does not need to be fit into the notches yet, we'll do that later.

Now to repeat these four fasteners on the opposite side.

Make sure all four nuts are still held in with tape. If any have fallen out, press them back into place.



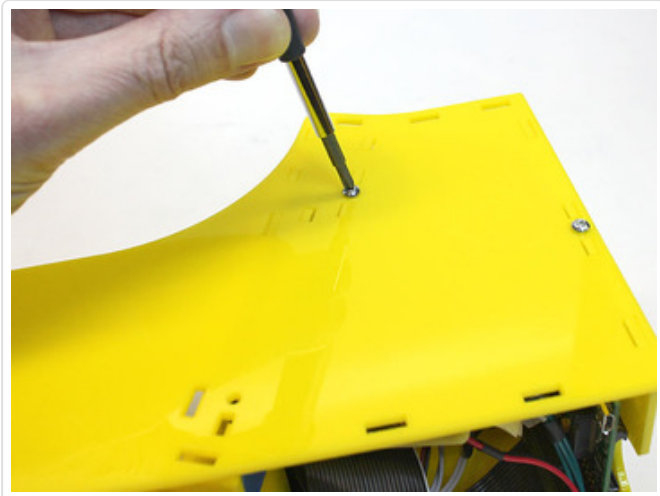
Now set the other side piece on top, roughly in position. I can *guarantee* the tabs won't all fit. That's okay, we'll align things one at a time…



Start with the base piece; this has the fewest interferences.

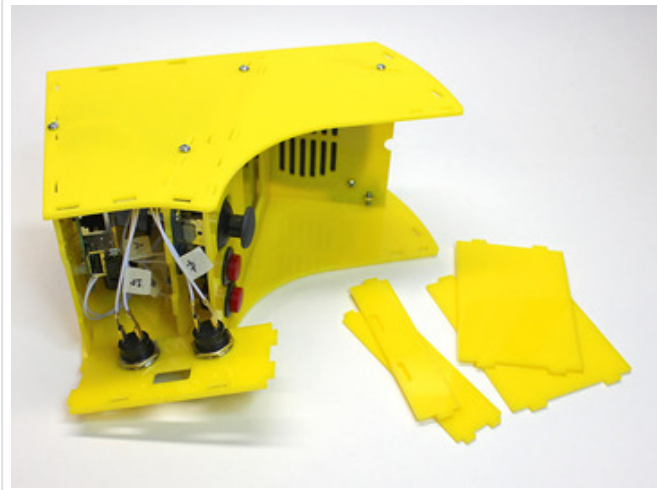
Get the tabs and hole aligned for this one part, then insert a screw. Do not crank it down…a loose fit is fine, just so it has a good hold on the screw.

Work your way from bottom to top, aligning the



tabs closest to each hole and then inserting a screw.

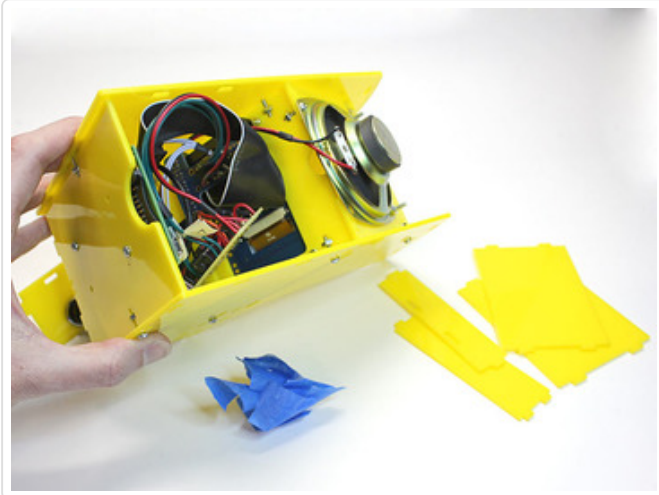
You may need to nudge each piece into position using the tip of the screwdriver through the slots, or with a finger from behind (when it can fit).



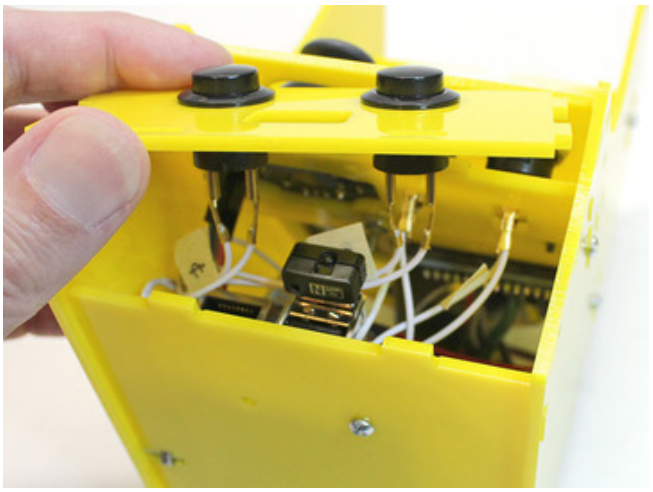
All four screws in place. Victory!

You can now remove the tape that was holding the nuts in place. Some of these will be deeply recessed; you might need tweezers to reach them. If you can't reach them, just leave them be… nobody will see them and they don't interfere with the system.

Now gather up the other four case parts…



Friendly reminder: this is your last chance to add



a USB mini WiFi adapter if you want! Plug it into the **top** USB slot. Leave the bottom slot open for a keyboard.

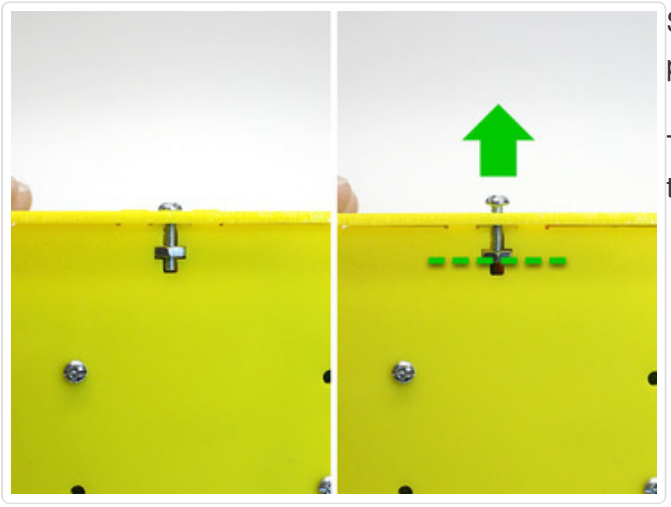
The case blocks access to the Ethernet port; WiFi is the only networking option.

Without WiFi, you can still load games into the system by inserting the SD card into a USB reader on your computer and moving ROM files to the /boot/advmame/rom folder.

Cabinet Part 4

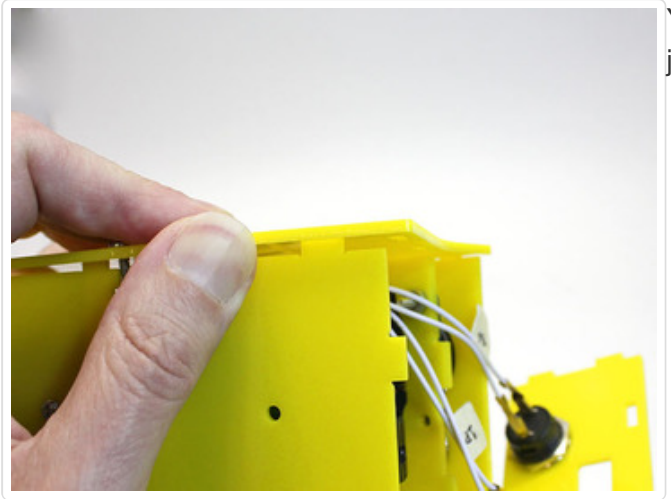
To fit the remaining pieces, it's necessary to ease up on some of the case screws…but only **slightly**, and **one at a time.**

If unscrewed too far, the nuts can fall off inside the case. Then you'll have to dismantle the case and go back several steps to rebuild everything. That's no fun, so be careful!



Start by loosening **one** of the screws on the base piece — either side, doesn't matter.

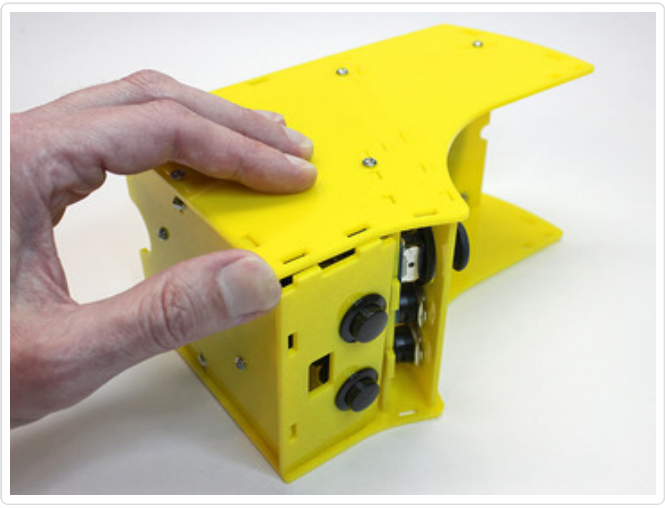
The tip of the screw should be flush with the face of the nut.



You should now be able to lift this edge of the case just a little.

Stuff the credit/start button wires inside, then slot this piece into the holes on the "table" side of the case. Pivot it upright…tabs on the base should fit into holes on the front…and finally pop the last two tabs into the "loose" side.

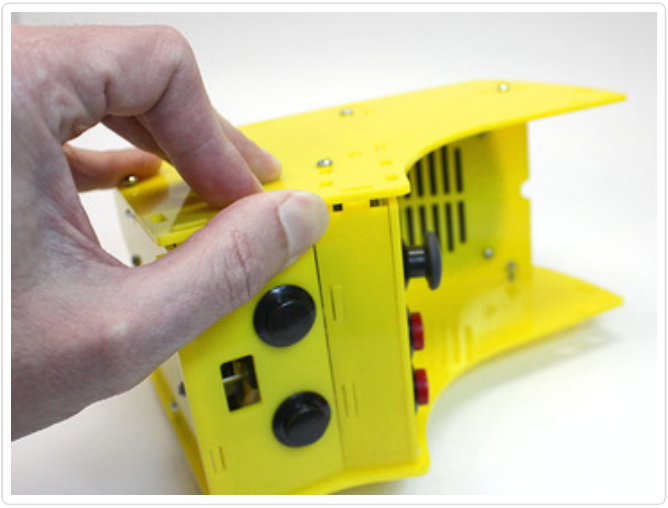
If you need a little extra working room, you can

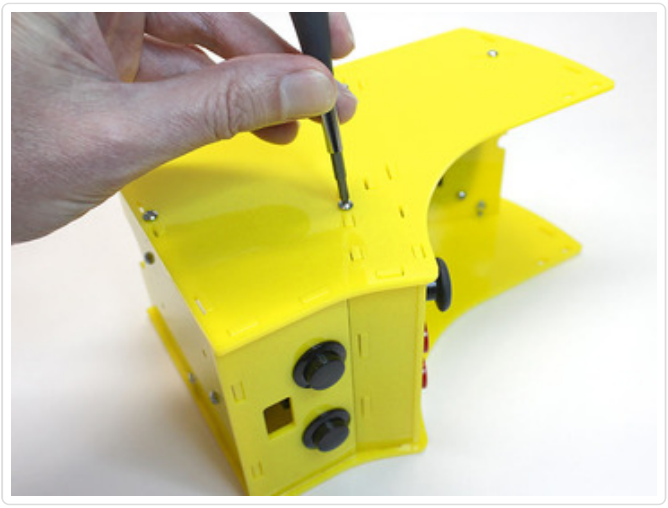


loosen the same screw on the opposite side, or the next screw up this side of the case…but don't take it as far, maybe unscrew half as much, or everything may come apart.

Once the button piece is in place, tighten the bottom side screw and loosen the next screw up this side…the one nearest the joystick. Loosen it a similar amount…screw tip flush with the nut face…no further!

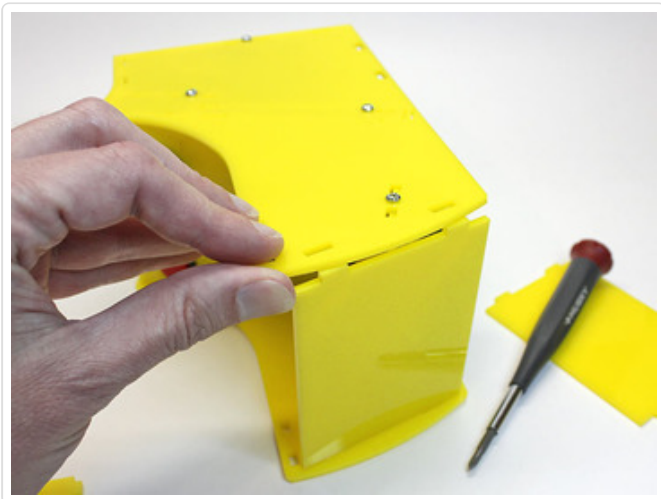
Install this fascia piece in a similar manner, then tighten the side screw back as it was.





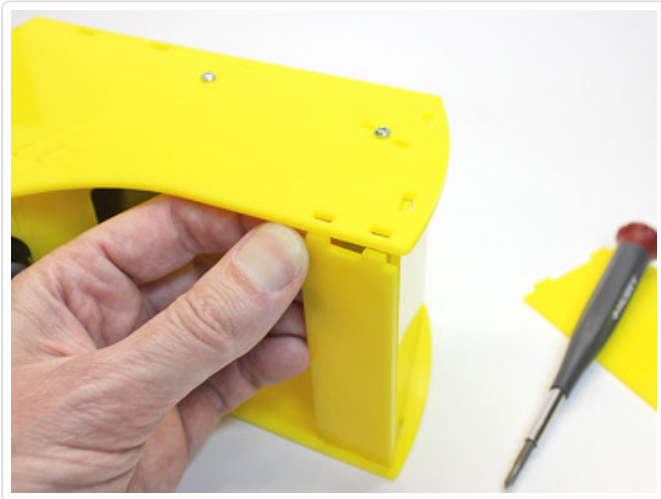
Loosen a side screw at the top. Same routine: screw tip flush with nut face, so you can flex this side up just a little bit.

Insert the “roof” of the cabinet. The roof and back pieces are similar in size, but the tabs are spaced



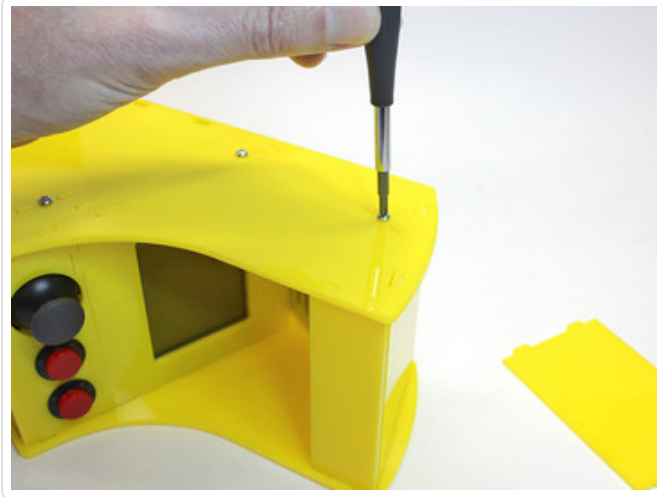
differently, so only the correct piece will fit.

As before, if you need a little extra working room, you can partway unscrew the opposite side.

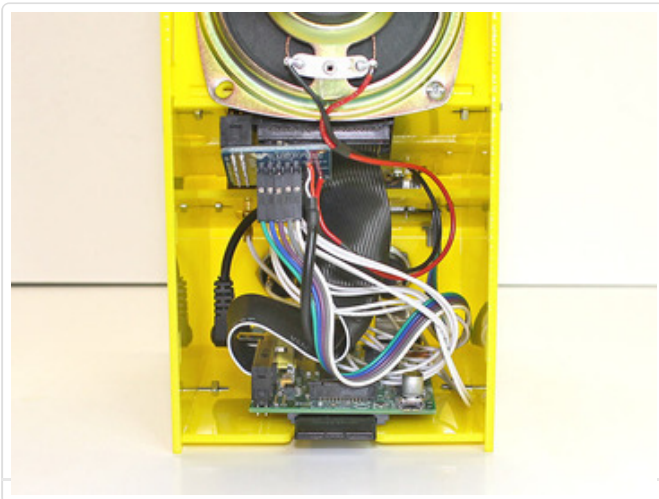


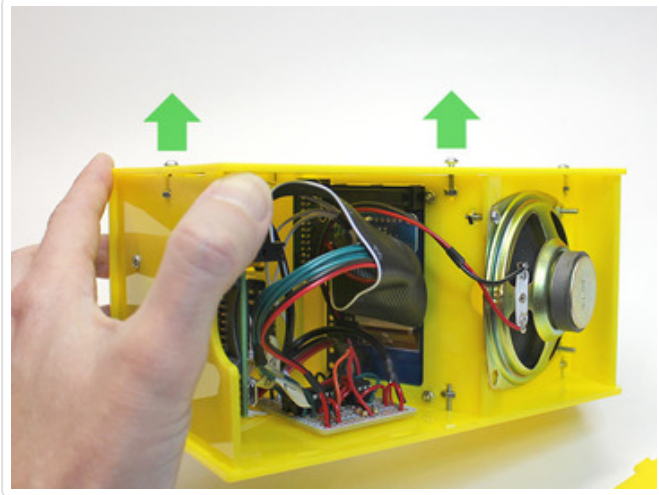
Slot the marquee into place, then re-tighten the top screw.

So close! Isn't it adorable?



Stuff the wires into the back of the case, being careful not to dislodge the Cupcade board.

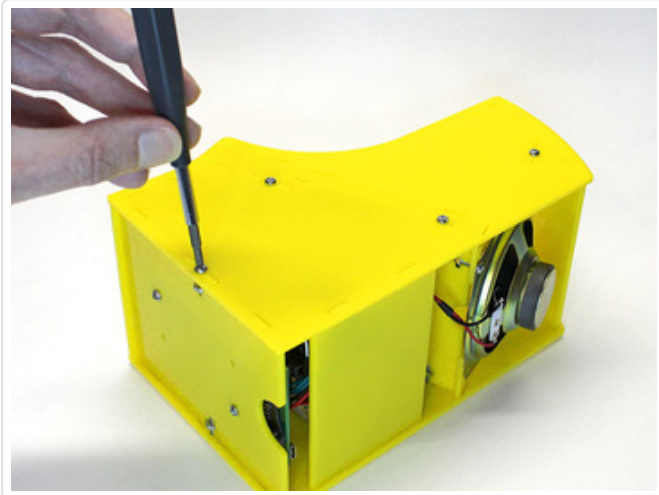


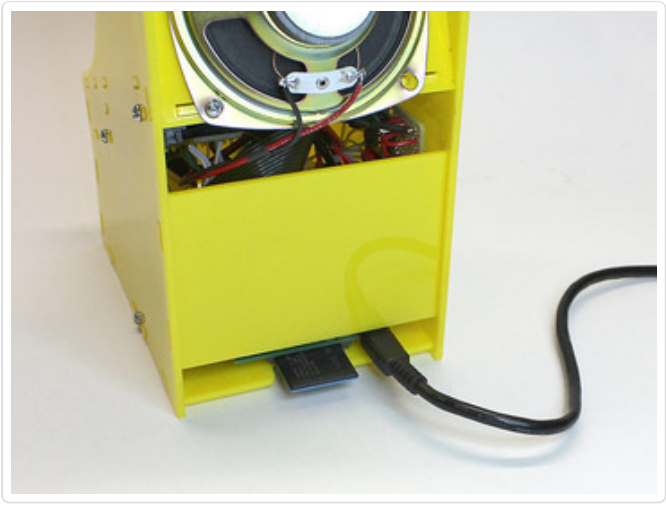


To install the back piece, you'll need to loosen two screws: the bottom and rear-most positions.

This piece pops into place like all the others.

When you're done, go around the whole case and make sure all the screws are snug. Not crazy tight…you don't want to crack the plastic…just tight enough to stay in place.





Insert the SD card, stand it up and connect power.

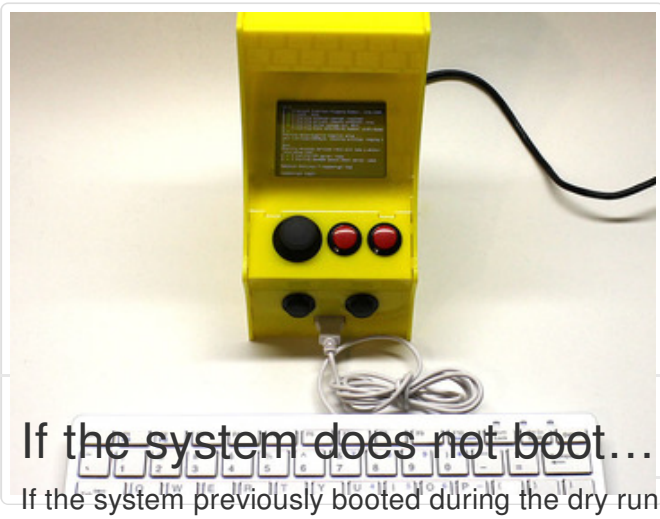
You can connect a USB keyboard at the front if you like. Some games with complex controls may require this, but for most classic games the basic stick-and-buttons are sufficient. The game selection menu and even shutdown can be accessed using combinations of these buttons.

Fire it up…see how it goes!

If everything works, that's great, **you're done!**

Remember: hold both the credit and start buttons for a few seconds to exit MAME and return to the game selection menu. Hold them again to exit the menu and shutdown the system.

Customize the case if you like. Stickers, paint markers or vinyl cut decals make it your own!



If the system does not boot…

If the system previously booted during the dry run but doesn't start this time around, remove the back of the case and check for the following:

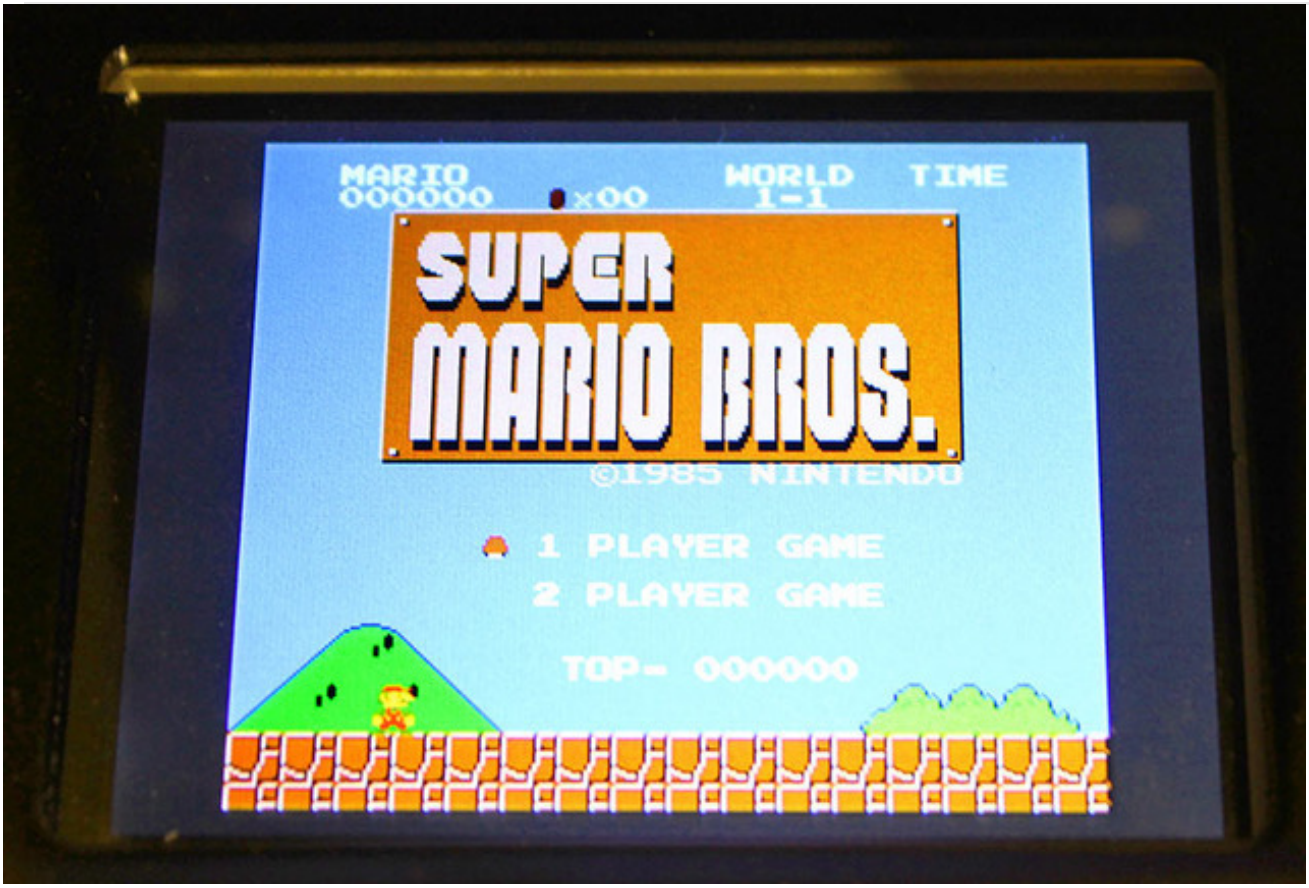
- Is the underside of the Cupcade board contacting anything metal?
- Is the Cupcade board properly aligned with the header on the back of the PiTFT?
- Have any of the wires broken off or come unplugged? How about the joystick? Are the wires still in place and in the correct positions?
- Is the SD card firmly seated in place, or has it wiggled loose?

My Cupcade rocks back and forth when stood up.

Loosen all 8 screws on the sides. A mere 1/4 to 1/2 turn or so…we're not dismantling the case, just realigning things.

Stand the case upright and press down gently on top, so all four corners make contact with the table. With your free hand, tighten each of the side screws back as it was.

NES Emulation



As of the 5-13-2014 Cupcade disk image, bonus NES emulation is provided via *FCE Ultra* (aka *fceu*).

This only works on Cupcades with a landscape (horizontal) screen. It will not work on vertical screens at all, period. It's just how the emulator is designed… it doesn't do video scaling and can't handle the narrow display.

The good news is that it's pixel-to-pixel scaled perfectly, so the display looks really good!

Click to Download Latest Cupcade (5-13-2014) with NES support

http://adafru.it/ds6

NES Controls

Something to keep in mind with NES games is that the controller (pictured above) has the “A” button (used for primary functions such as fire or jump) on the *right side.* If you spent a lot of time with an original NES and want to preserve that “muscle memory” with your Cupcade, then you may want to assemble your kit with the A button on the right.

However…many (but not all) arcade games favored an A/primary button on the *left.* This is not a hard-and-fast rule, because every arcade game had its own distinctive controls, so we have not set up the emulators to handle the buttons differently…the primary/secondary button mapping as provded is the same for both emulators. If you don’t have that muscle memory for the specific games, it won’t make a big difference.

If you really want to change this, there are a couple ways to do it…

First, if you want to change it globally for all arcade games, edit the file /boot/advmame/advmame.rc.common, swapping the button1 and button2 mapping:

```
input_map[p1_button2] keyboard[0,lcontrol] or keyboard[0,z]
input_map[p1_button1] keyboard[0,lalt]     or keyboard[0,x]
```

Or if you want to change it only for specific games (leaving the rest with the NES layout), copy the

original button1/button2 lines, swap the “1” and “2” on the copy and then *add the game name and a forward slash(/) at the start of those lines:*

```
xevious/input_map[p1_button2] keyboard[0,lcontrol] or keyboard[0,z]
xevious/input_map[p1_button1] keyboard[0,lalt]     or keyboard[0,x]
```

Then assemble your Cupcade case with the A button on the right, a la NES. It’s done this way because it’s easier to edit the *advmame* configuration than *fceu* (which uses a binary file, not human-readable).

Details and Errata

System Setup

If you're interested in setting up a system manually…and for the sake of documenting the process… we'll summarize the steps involved.

Rather than use a build system to generate a pristine bootable card, we worked from a booted system, installing packages and making configuration changes, then did our best to "stuff the genie back in the bottle" and make it act like a first-boot system. A bit uncouth, but it works.

Burning your own SD card

Kit purchasers will need to burn this file onto the 4G SD card.

Begin by downloading our pre-built Linux disk image for the Cupcade. This saves a *whole lot* of configuration steps!

Click to Download Latest Cupcade (5-13-2014) with NES support

http://adafru.it/ds6

Click to Download Original Cupcade Disk Image (4-4-2014)

http://adafru.it/dg7

This is a big file (about 840 megabytes) and will take a while to transfer.

After downloading, the disk image needs to be installed on an SD card (4GB or larger). If you're new to this, the process is explained in **Adafruit's Raspberry Pi Lesson 1** (http://adafru.it/aWq) and/or check out this tutorial on how to burn SD cards with Raspbian images (http://adafru.it/aMW)

Core OS and TFT Support

Creating our own OS image began with a current, stock Raspbian SD image, apt-get update and upgrade applied, atop which several packages are then installed to support the PiTFT. This includes the latest DMA-enabled display driver. The necessary packages and installation procedures are all documented in the Adafruit PiTFT guide (http://adafru.it/dcX). X11 setup and touchscreen calibration are not performed, since all the emulators run full screen and do not use the touch feature.

Normally one would set the environment variables FRAMEBUFFER and SDL_FBDEV in /etc/profile,

but that's not necessary here. These values are instead set by inittab each time the emulator is launched.

The TFT interface is clocked at 80 MHz; technically out of spec, but it's worked for us so far. This is configured in /etc/modprobe.d/adafruit.conf:

```
options fbtft_device name=adafruitts frequency=80000000 fps=60 rotate=90
```

The last parameter sets the screen rotation. This can be edited manually, or there's a feature of the game selection menu to make this easier for laypersons: connect a keyboard and press SHIFT+R. The program will make the necessary change and reboot immediately.

Additional steps are explained in the aforementioned PiTFT guide. These include additions to /etc/modules and enabling the Terminus 12x6 console font. /etc/modules includes the following four lines:

```
snd-bcm2835
spi-bcm2708
fbtft_device
snd_pcm_oss
```

Some system configuration changes have been made via raspi-config:

- Internationalization configured for United States English locale and keyboard.
- Hostname is set to cupcade.
- ssh is enabled.
- Audio is forced to 3.5mm (headphone jack) output.
- GPU memory reduced to 16 MB.
- System is overclocked to 900 MHz.

The default account password "raspberry" is left as-is. ALSA gain is set to 0 dB.

A few additional packages are then installed:

```
sudo apt-get install netatalk ncurses-dev libexpat1-dev
```

The first enables Zeroconf (aka Bonjour), making the system accessible on the local network as cupcade.local. The latter two are for compiling the retrogame utility (explained later).

The boot partition has been resized to about 400 megabytes to allow ample space for ROM files and other configuration data.

AdvanceMAME

There are a few different version of MAME (Multiple Arcade Machine Emulator) for Raspberry Pi. Some use OpenGL for the display, making them incompatible with the PiTFT. This is why we suggest Shea Silverman's build of *AdvanceMAME* for Raspberry Pi. It can be downloaded and installed using the following commands:

```
wget http://sheasilverman.com/rpi/raspbian/debs/advancemame_1.2-1_armhf.deb
sudo dpkg --force-overwrite -i advancemame_1.2-1_armhf.deb
```

Normally advmame needs to be run once to set up paths and create a default configuration file. However, we suggest setting it up so it uses the /boot/advmame directory instead, so these files can be accessed by plugging the card into a regular PC.

Update: as of 5/13/2014, we suggest version 0.94. This earlier release performs better with the Raspberry Pi's limited resources.

Games go in the /boot/advmame/rom directory. Also in the /boot/advmame directory are *three* configuration files:

1. advmame.rc.portrait (contains specific options for vertical display)
2. advmame.rc.landscape (contains specific options for horizontal display)
3. advmame.rc.common (contains settings used regardless of screen type)

When run, one of these configuration files is passed as a parameter to advmame, e.g.:

```
advmame -cfg /boot/advmame.rc.portrait gamename
```

The user normally doesn't see this…it's hidden away by the menu program used to launch games. It takes care of specifying the correct configuration file based on the rotate value in /etc/modprobe.d/adafruit.conf.

A dummy /root/.advance/advmame.rc file was added (0 bytes). advmame, when spawned by a root-owned process, won't run without it.

Finally, the XML-formatted game list is generated:

```
advmame -listxml > /boot/advmame/advmame.xml
```

This provides human-readable descriptions for each game (rather than just filenames). This file has been pre-generated on the SD image, in the /boot/advmame directory. The normal advmame folder (home/pi/.advance) is not used; everything's in /boot.

Normally there's some additional setup required for video modes (using the advv utility), but the provided advmame.rc files are pre-configured for the PiTFT.

ROM files for fceu (FCE Ultra, a NES emulator) can go in /boot/fceu/rom

Retrogame and Gamera

retrogame is the code (run as a background task) that translates GPIO input (from our joystick and buttons) into virtual keyboard presses that emulators can understand. This lets us use a plain vanilla advmame; no additional code needs to be hacked in for GPIO support. It can be downloaded from Github:

```
wget https://github.com/adafruit/Adafruit-Retrogame/archive/master.zip
```

Unzip the archive and such. The resulting directory contains a pre-built version of retrogame, but it was designed for an earlier, simpler gaming project. A special version needs to be compiled for the Cupcade:

```
rm retrogame
make CFLAGS=-DCUPCADE
sudo make install
```

The retrogame source code already has the Cupcade-specific key bindings built in, they just need to be enabled this way.

UPDATE: retrogame now detects Cupcade automatically; CFLAGS=-DCUPCADE is not required if you download the source from the link above. The pre-built SD image is still working from the older code though.

Because it accesses hardware directory, retrogame needs to be run as root (and in the background, so other emulators can then be launched):

```
sudo retrogame &
```

Our SD image references retrogame in /etc/init.d/rc.local, so the root and background aspects are taken care of automatically at startup.

gamera (Game Rom Aggregator) is the game-selection utility. Visually it's nothing special — text-based, using the *ncurses* library — but it does the job. The code scans the rom folder, correlating items there to the contents of the advmame.xml file, then displays a list which can be navigated

using the controls (or a keyboard). Upon selecting a game, it launches advmame, passing it the correct configuration file for the screen layout, plus the game name.

You probably don't need to recompile gamera.c unless files or directories have moved around. But if you do, this is why the ncurses-dev and libexpat1-dev packages were previously installed with apt-get.

gamera is spawned via inittab, so it takes over the normal console login. To access a login prompt when the menu is running, press Alt+F3 or Alt+F5. The relevant lines in /etc/inittab are:

```
1:2345:respawn:sudo -i FRAMEBUFFER=/dev/fb1 openvt -f -c 1 -w /home/pi/retrogame/gamera
#1:2345:respawn:/sbin/getty --noclear 38400 tty1
```

(The second line is simply the original inittab getty line for tty1, commented out for the time being.)

gamera behaves a little differently when started by root rather than pi (or other regular user). For root, the ESC key will shut down the system. Since gamera is run at startup from /etc/inittab (as root), this allows the system to be shutdown without plugging in a keyboard (hold down the credit and start buttons for a couple seconds).

Unlike the other files that are stored in the /boot partition, the retrogame and gamera programs and source code reside in /usr/local/src/retrogame. Compiling code in the /boot partition just leads to file access and ownership gas pains. After compiling, sudo make install will place the executables in the required location (/usr/local/bin).

Custom Cabinet Art

Looking to create custom artwork for the sides, marquee and display/control bezels? Here's an artwork file (PDF format) you can use as a template:

cupcade-cab-art.pdf

http://adafru.it/dD9

If making vinyl stickers for the sides, you'll probably want holes for just the screws, not every slot. It's all there for reference though and you can delete the bits you're not using. For the marquee and top/front surfaces, stickers might fare better without the assembly tabs.

Blue outlines show the areas obscured by the buttons or screw heads, and also the ~3mm lip along the front edge of the control bezel. *(I apologize for the screw heads, but they had to go somewhere, and I took great pains to minimize the number.)*

For a smoother appearance, orient and align the screen and joystick bezels to match your setup,

then make a single combined sticker with cutouts for the screen and controls. You could even keep going and make this wrap all the way down the front in a single unbroken piece, though it might make assembling the case extra tricky.

Troubleshooting!

The system won't boot during the "dry run" test!

Carefully unplug the Cupcade interface board from behind the PiTFT and try booting again. Does it work now? If so, the problem is usually with the new parts.

- Was the board properly aligned with the header on the PiTFT? It's easy to get this off by one.
- Are any of the button quick-connects making accidental contact with each other?
- Is the joystick connected? Cupcade won't boot without it!
- Are the four wires connected to the proper pins in the correct order on the joystick? Even a single mislaid wire here can prevent the Pi from booting!
- Examine the joystick closely, specifically the horizontal and vertical potentiometer dials on the sides. When you move the stick around, can you see both dials turning with it? If not, the joystick may be faulty.

Some of the buttons/joystick work but not all!

Check if you have all the connections solid and going to the right location. The joystick socket connectors can slip off if yanked!

The system used to boot fine, but stopped working!

You can download a fresh SD card image on the "Details and Errata" page. Write this to the card as you would any other Raspberry Pi OS image.

There are a couple of reasons the card may have been corrupted:

- System was not properly shut down. Always issue an orderly shutdown by holding down both the Credit and Start buttons from the game list.
- There are some reports that overclocking the Raspberry Pi runs a small risk of corrupting the card contents. The games really do play noticeably better with overclocking, so we're taking that chance; it's not that hard to prepare a new card if it becomes necessary (just be sure to keep a backup of your ROM files).

What games are known to work/not work? Any caveats?

Here's some games we tried out with success:

Vertical:

- Arkanoid (arkanoid.zip)
- Dig Dug (digdug.zip)
- Donkey Kong (dkong.zip)
- Donkey Kong Jr (dkongjr.zip)
- Frogger (frogger.zip)
- Galaga (galaga.zip)
- Ms Pacman (mspacman.zip)

- Scramble (scramble.zip)
- Vanguard (vanguard.zip)
- Xevious (xevious.zip)
- Zaxxon (zaxxon.zip) - sound samples don't work by default, see notes below.

Horizontal:

- Tetris (atetris.zip) - 336x240 game so 8 pixels are cut off but its not noticeable
- Joust (joust.zip)

Here's some games with issues:

- Altered Beast (altbeast.zip) - slow!
- Gauntlet (gauntlet.zip) 336x240 so a few pixels cut off. Sluggish but playable on original 4-4-2014 image, much better on new 5-13-2014 image!
- Pole Position 2 (polepos2.zip) - controls dont work
- Shinobi - use shinobi2.zip (set 2 system 16b) not shinobi1.zip. sluggish but playable.
- Golden Axe (goldnaxe.zip) - slow!

Known Issues

- Sound is "warbly" when games first start (e.g. *Pac-Man* music). It will clear up after a few seconds. Emulation is demanding for the Pi and it takes a moment for the newly-loaded code and data to be cached for efficiency.
- 68000-based games are pretty intensive and will run sluggishly even on an overclocked Raspberry Pi. This includes *Altered Beast, Gauntlet* and *Namco Classics Collection Volumes I & II.*
- Scanning the ROM folder is time-consuming. There's a massive XML file that must be skimmed in full, even if you have just one or a handful of ROM files. If this is annoying, you can delete /boot/advmame/advmame.xml, but games will show only filenames instead of nice descriptions.
- *Tempest* (and probably other vertical vector games): improper scaling on vertical (portrait) screens; it's too small. This appears to be a bug in advmame making assumptions about the screen aspect ratio. A semi-workaround is to tell the system you have a horizontal screen (despite being positioned vertically) and changing display_ror to yes in /boot/advmame/advmame.rc.landscape. The downside is that the console (and the menu program) will be turned sideways.

Fixing Sound Samples

Some games (such as Zaxxon) can't generate audio on the fly; they require WAV audio samples. The current Cupcade disk image doesn't handle this yet, but it's easy to fix in the meantime:

1. Create folder /boot/advmame/sample
2. Edit /boot/advmame/advmame.rc.common, adding this line:

3. Drop your unzipped samples folder (e.g. “zaxxon”) into /boot/advmame/sample
4. Boot Cupcade from this card and play! Pew pew pew!

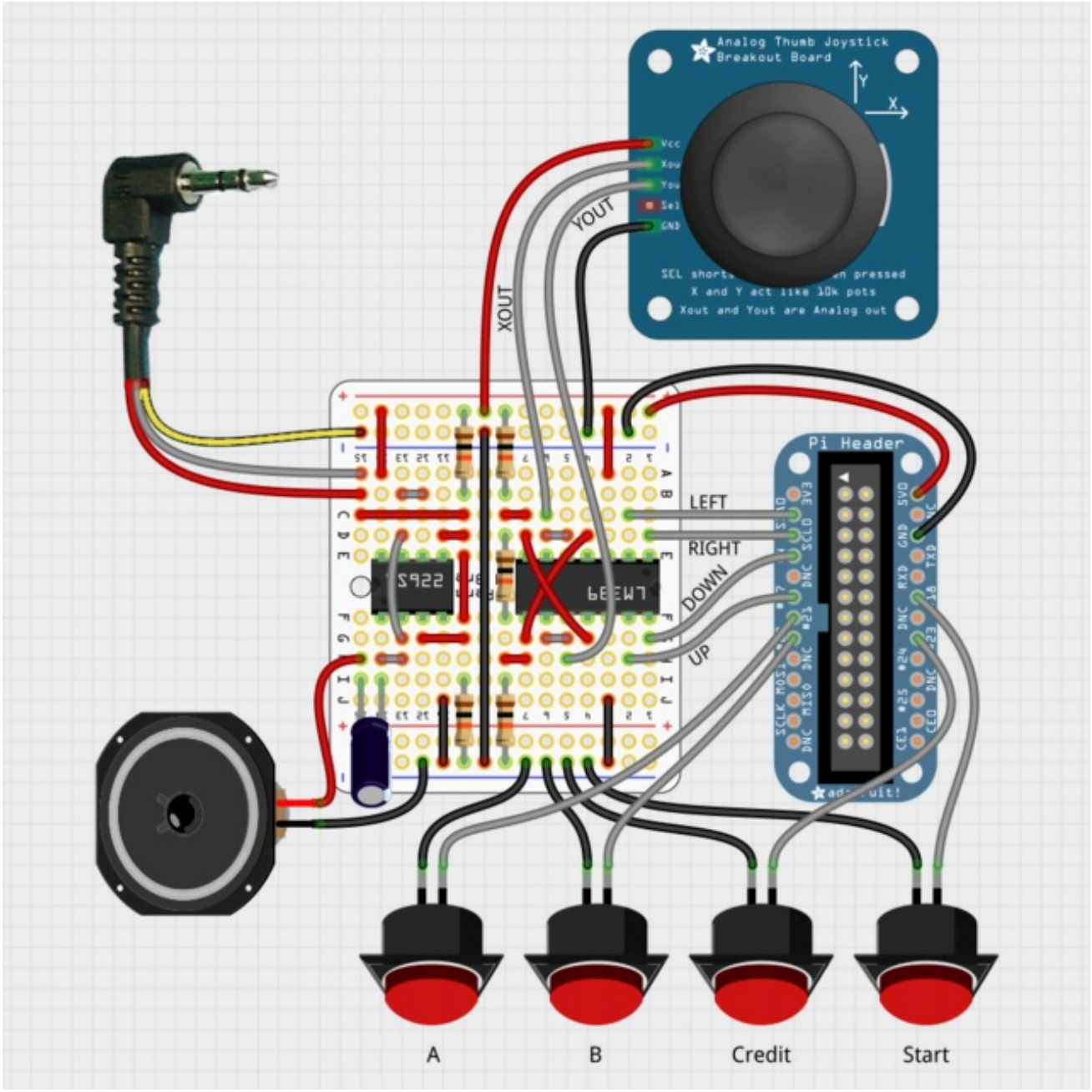
Beta Board Part 1

These directions apply only to the BETA Cupcade units. Most users can refer to the simplified final design on the “Interface Board” page.

It’s a tight squeeze, but everything *just* fits on a 1/4 size Perma-Proto board!

The Perma-Proto circuit is neatly divided into two sections. The left side is a basic audio amplifier for the speaker. The right side interfaces the analog joystick to the Raspberry Pi’s digital-only GPIO. The line down the middle connects the two ground rails across the board.

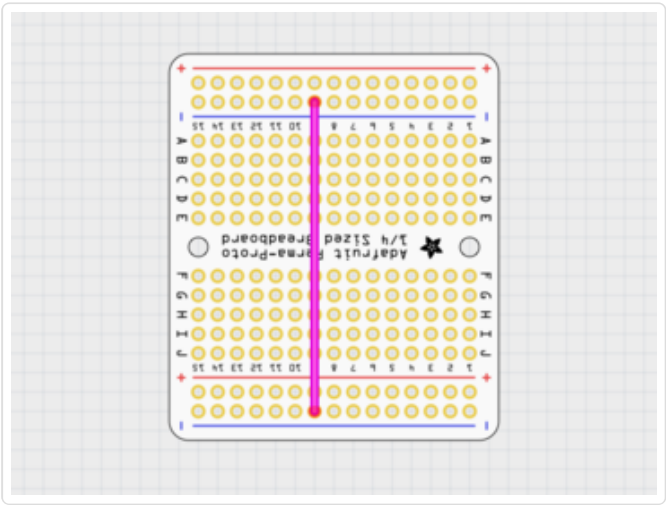
This diagram shows a rectangular Pi Cobbler; we’ll actually be using a “T” cobbler. Same functionality, same pins, just a different shape than shown here.



The Perma-Proto is turned around, with the labels upside-down. It's fine, this orientation just makes our circuit easier to lay out.

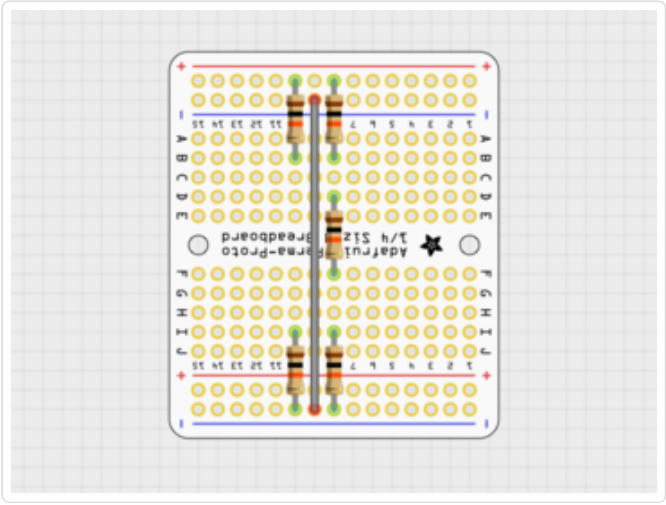
Cut a wire a little over 2 inches long and strip the ends. Run this along **row 9** between the two **ground** (blue) traces.

Aside from joining the ground traces, this provides a nice visual separation of the two unrelated parts



of the circuit. It's okay if this wire is a little longer and doesn't lie flat; strictly a visual thing.

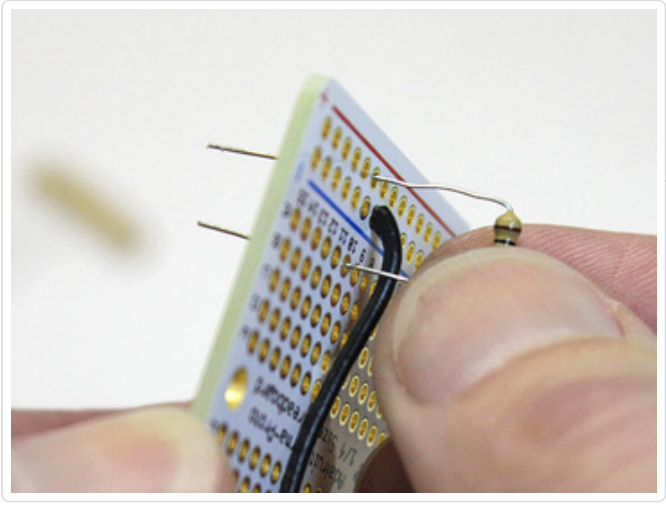
Make certain you're connecting the **ground** (blue) traces, not one or both of the positive (red) traces.



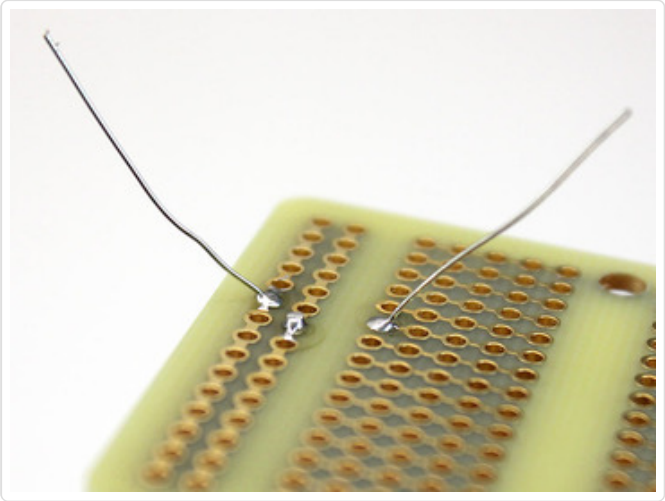
Install five **10K resistors** (brown-black-orange) in the positions shown, immediately to the left and right of the ground wire. Two go on the left (audio) side, three on the right (joystick) side.

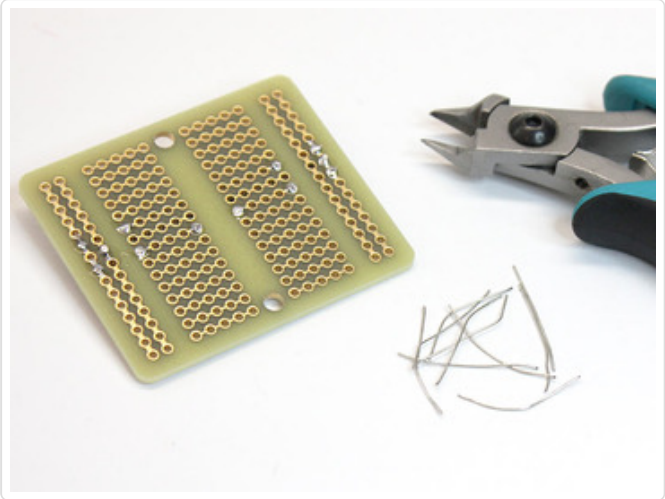
The two resistors at the **top** connect to the **positive** (red) trace, while the two at the **bottom** connect to **ground** (blue). The final resistor bridges the center gap on the right side.

Bend the resistor legs so they're parallel like staples, insert them into the Perma-Proto board so they're sitting flush and then bend the legs on the back outward to hold them in place for soldering.



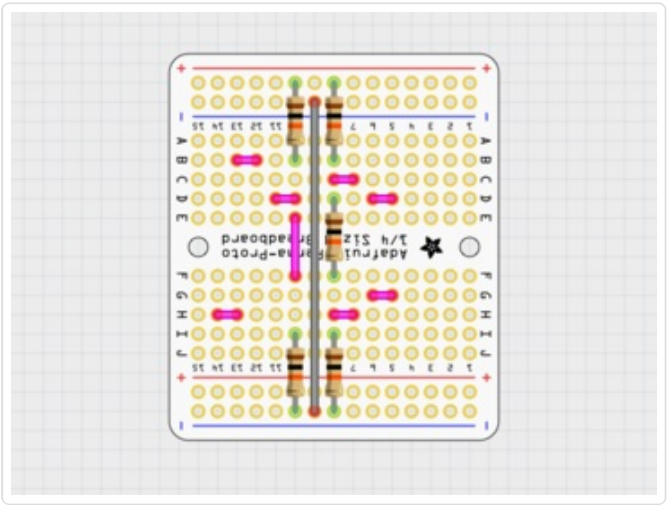
After soldering, the legs can be trimmed close to the board. **Save these trimmings** for the next step — they make handy jumpers!

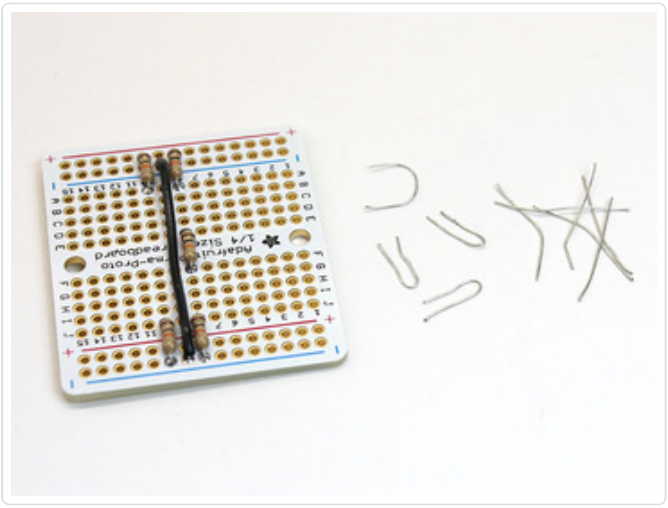




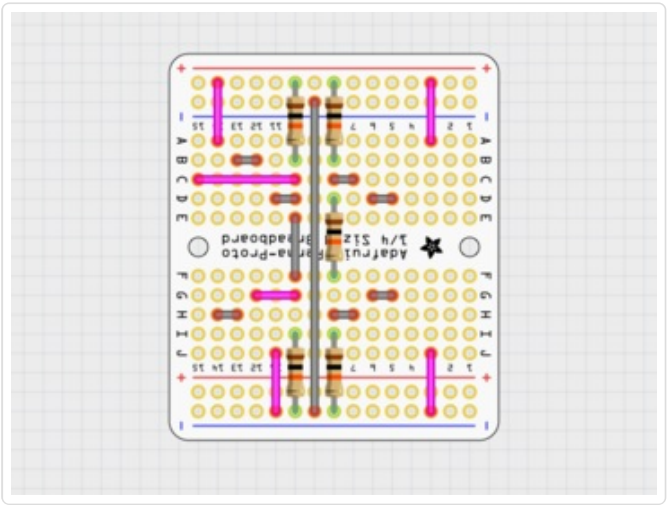
Eight short jumpers are now installed in the positions shown in pink. Most of these are very small, joining adjacent pads. We can use the un-insulated clippings from the prior step because no other conductive traces are crossed.

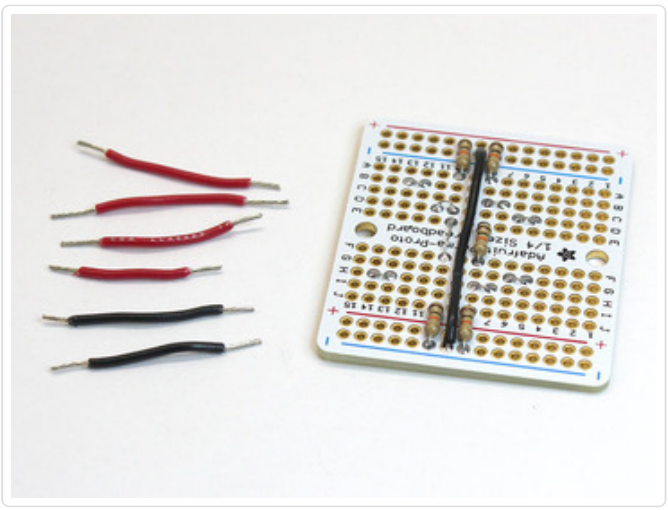
Install these similarly to the resistors: bend each jumper into a small "staple" shape, insert it through the correct holes (triple-check!) and bend the legs outward to hold for soldering.

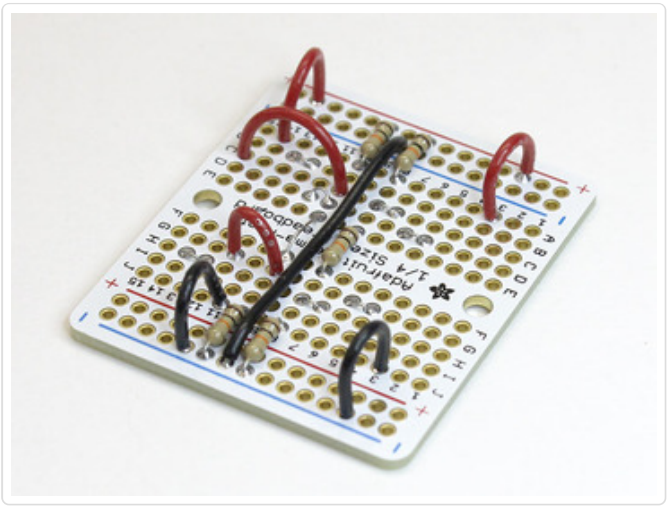


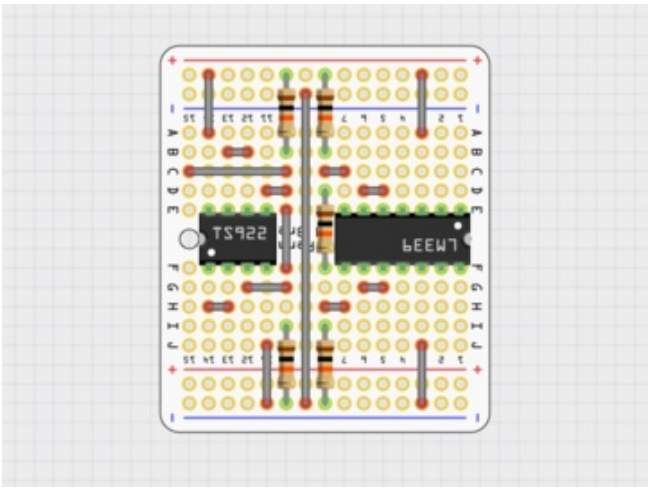


Six more jumpers are now installed as shown in pink (prior steps are gray). Because these jumpers cross other traces, they should be cut from **insulated** wire, do not use the resistor clippings for this step!





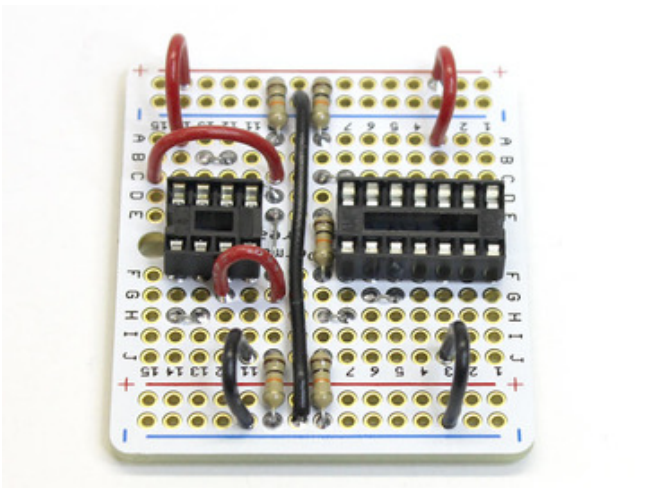




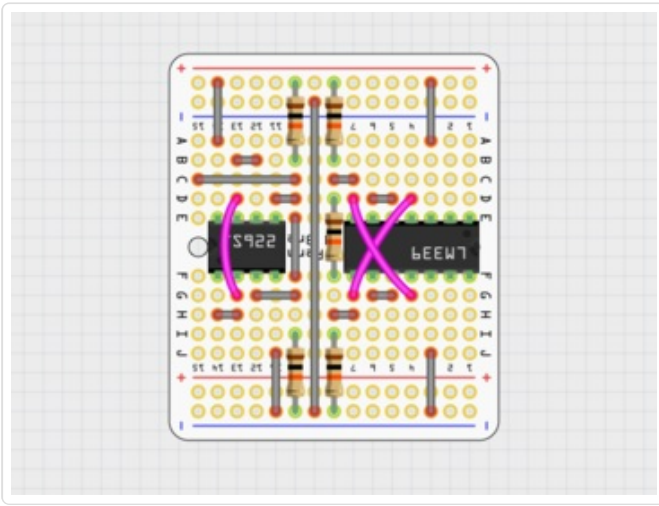
This diagram shows where the TS922 amplifier and LM339 comparator ICs will be installed later. In their place for now we'll install **sockets,** then add the chips after everything else is complete.

Make sure you get the 8-pin socket in the correct position — it's inset by one row, not right at the edge of the board.

Note that the chips will be installed "back to back," with pin 1 facing opposite edges of the board; they're not oriented the same way.

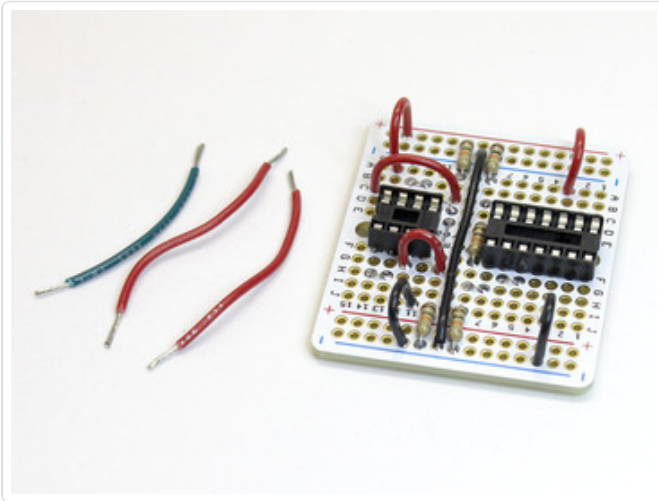


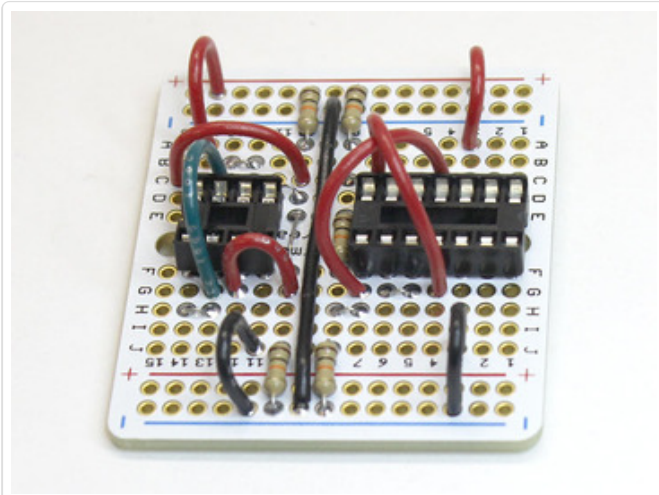
When soldering sockets, use only enough solder to make a good connection between the pins and vias. Don't keep adding solder…you'll fill the socket holes and won't be able to insert a chip!

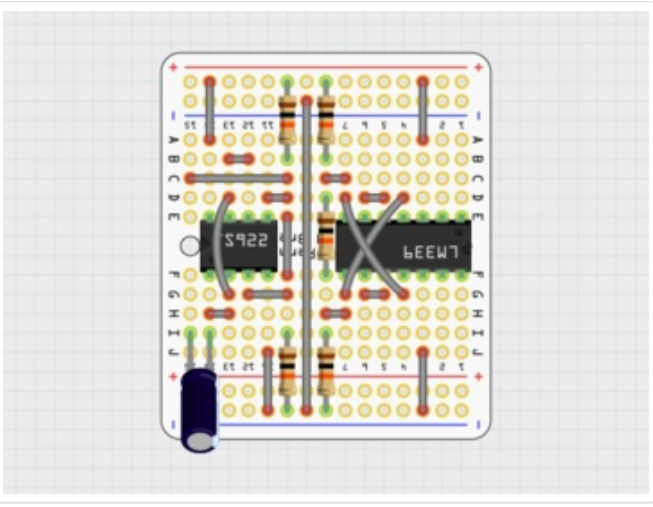


Three more jumpers are installed, cut from insulated wire.

These wires cross over the IC sockets. Do not follow a straight shot as depicted in the diagram — that's just to make the connection points more clear. The wires should actually be a little longer and will get bent *around* the sockets, so a chip can still be inserted.

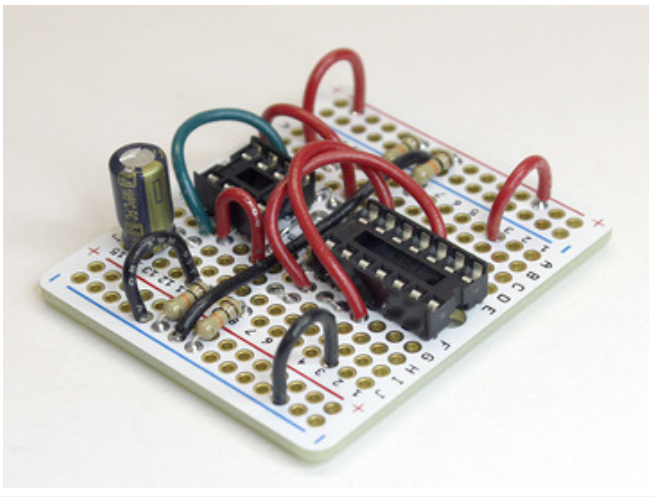






Install the **100 µF capacitor** in the position shown.

In the photo, notice the wires now bent around the sockets. Don’t add the chips yet, we’ll do that later, after the rest of the soldering.



This is a good time to pause and double-check your work.

- Do your wires and components match the layout precisely as shown? Count the number of wires and components and the spaces between them.
- Are there any cold solder joints or shorted pads on the back of the board? Now's the time to repair them.
- Are all the wire ends trimmed?

If something is installed wrong, *at best* the circuit won't work as expected. At worst, the Raspberry Pi won't boot or could even become **permanently damaged!**

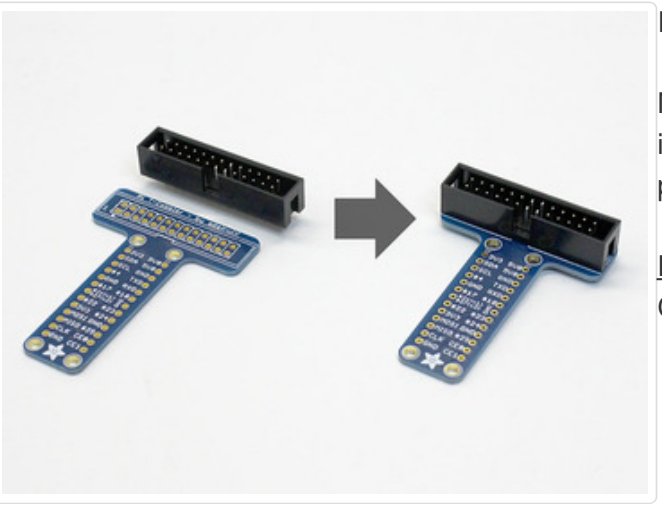
The circuit is 5 Volts but connects to the Raspberry Pi's 3.3V GPIO pins. Won't this fry the board?

The LM339 comparator outputs are *open drain.* When active, they connect to ground (0V). When inactive, they "float" (aren't connected to anything) and the Pi's own pull-up resistors register this as a 3.3V "high" signal. The unregulated 5 Volts (which we really need for the amplifier) never gets back to the Pi…assuming it's all assembled correctly, so please double-check everything!

Beta Board Part 2

These directions apply only to the BETA Cupcade units. Most users can refer to the simplified final design on the “Interface Board” page.

This page is mostly prep work: measuring, cutting and stripping wires, making labels and bits of heat-shrink tube. Doing these steps carefully should make the next steps less troublesome.

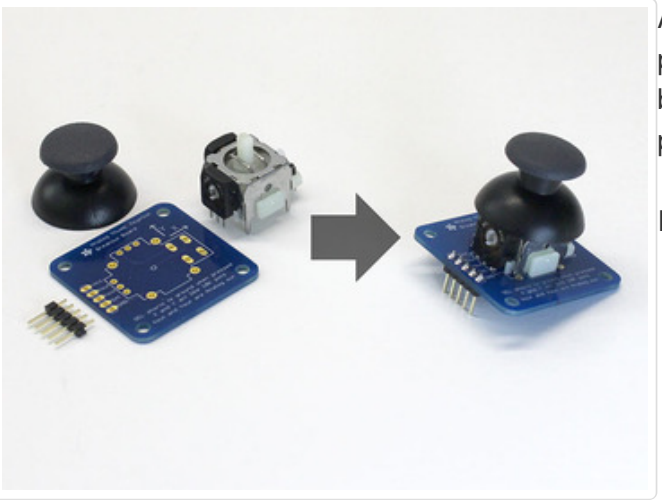


Install the male (pin) header on the Pi T-Cobbler.

Make sure the “key” (notch) is on the correct side; it’s marked on the silkscreen. It should face all the pin labels.

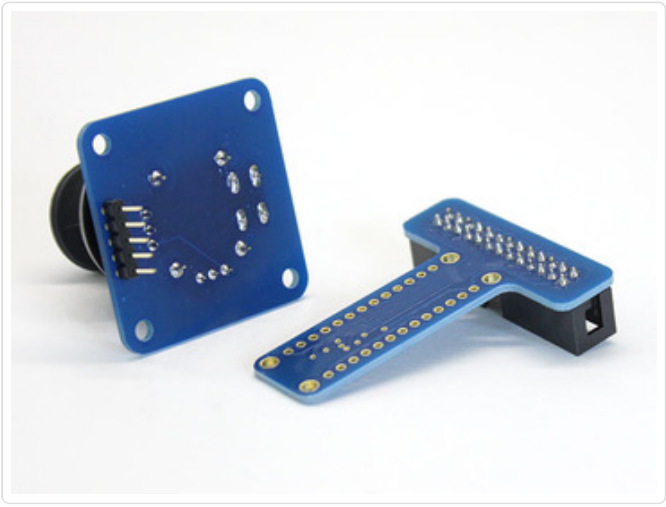
Do not install any other pin headers on the Cobbler! We’ll be soldering wires directly later.

Make sure to solder the socket on correctly. The socket goes on the same side as the text and the notch is pointed toward the Adafruit logo.

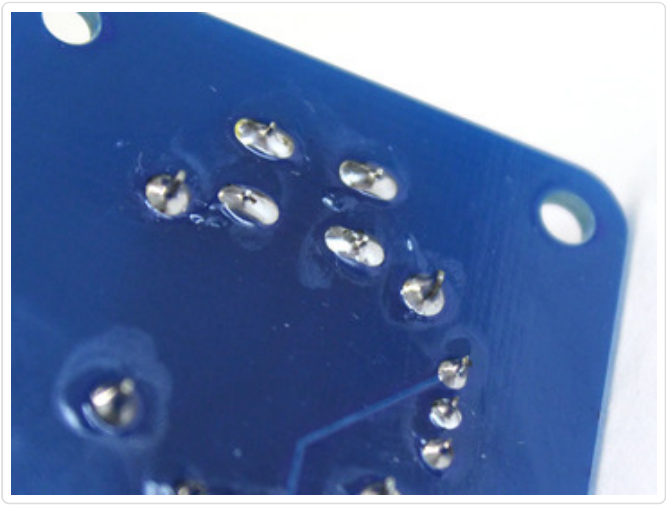


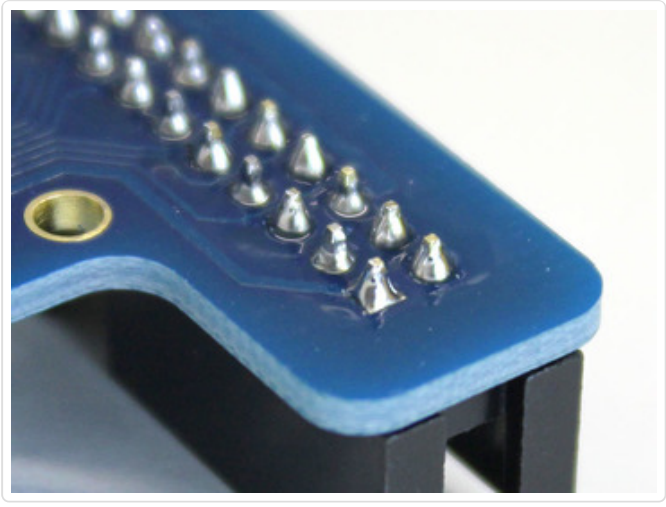
Assemble the analog thumb joystick. Some of the pins may be a little bent from shipping and need to be nudged into place. The “hat” simply presses into place.

Install a 5-pin header from below.



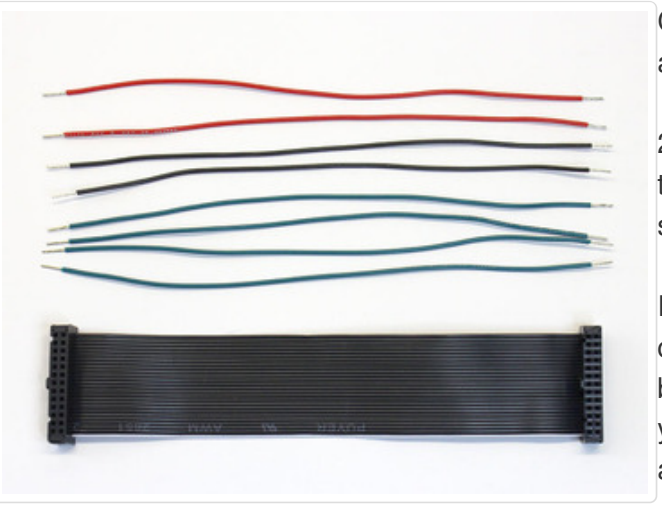
Make sure all your solder connections are smooth and clean, like tiny Hershey's Kisses®. There should be no solder balls or bridges, no gaps between pins and their corresponding pads.





A whole mess of wires need to be cut to about six inches long. You don't need a ruler for this…the ribbon cable is a perfect size. Unplug it from the Pi and PiTFT and use it for reference.

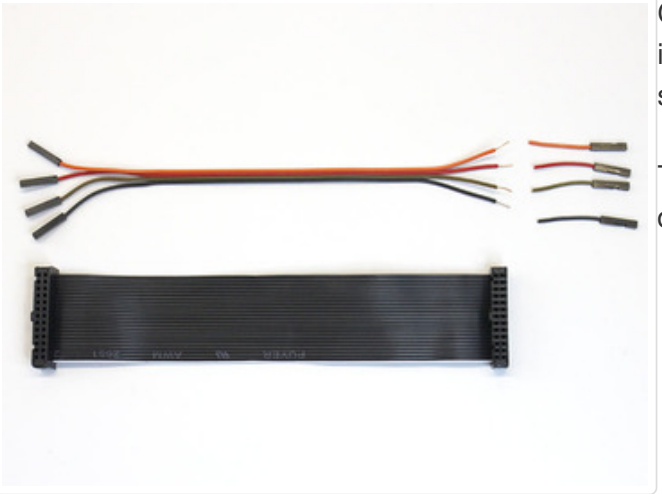
This isn’t rocket surgery, don’t worry if wires aren’t exactly the right length. Six-*ish* inches is fine!



Cut eight (8) wires about six inches long and strip about 1/4" of insulation from both ends.

24 gauge stranded wire is ideal. A little thicker or thinner is okay, as is solid-core wire…stranded is simply more flexible.

If you have some different colors of wire, you can color-code these. Here we’ve cut two each red and black (for power), and four green (for signals). If you don’t have different colors, that’s okay, we’ll also be labeling them in a bit.

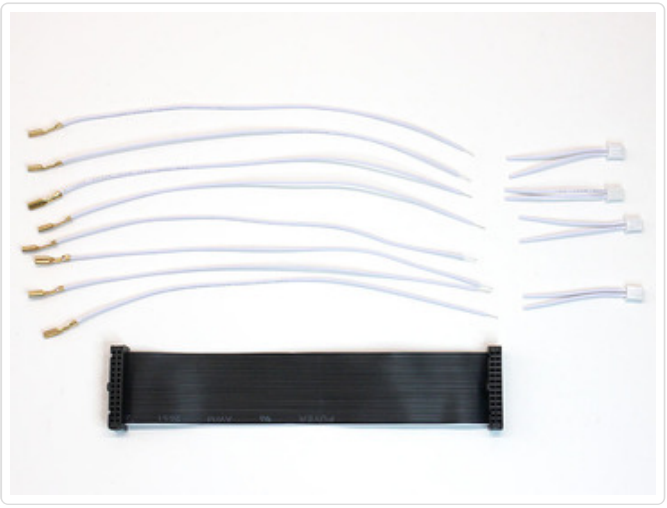


Clip one end off the 4-conductor jumper cable so it's a similar length. Separate the wires a little and strip about 1/4" of insulation.

The wire colors are random; don’t worry if yours don’t match the photo.

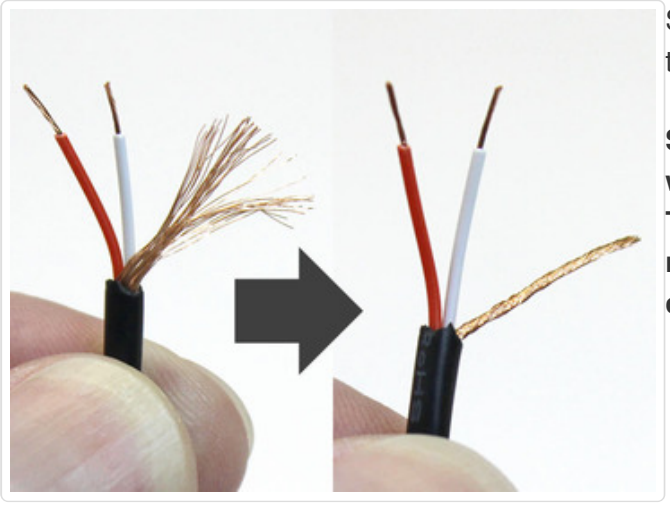
Clip the 2-pin plastic plugs off the button quick-connects. You should then have eight (8) wires with a metal spade on one end. Strip a little insulation from the other end.

Something about the composition of these wires is strange and slightly brittle, and you may have trouble with your first couple of tries stripping them. No problem…clip these a little long to start, and if the stripping goes badly, you can trim it off and try again.



Cut the audio cable so it's a little longer than the ribbon cable.

Strip about 1" of the *outer jacket only* to reveal the three wires inside: two are insulated (red and either white or black) and the third is bare copper.



Strip 1/4" insulation from the two inner wires. Twist the stranded copper wire into a tidy bundle.

Some headphone cables have red and black wires rather than the red and white shown here. This is OK, it all works the same. Red is the right channel, black or white is the left channel, copper is ground.

Arts & Crafts Time!

Labeling the wires will make later steps much, *much* easier!

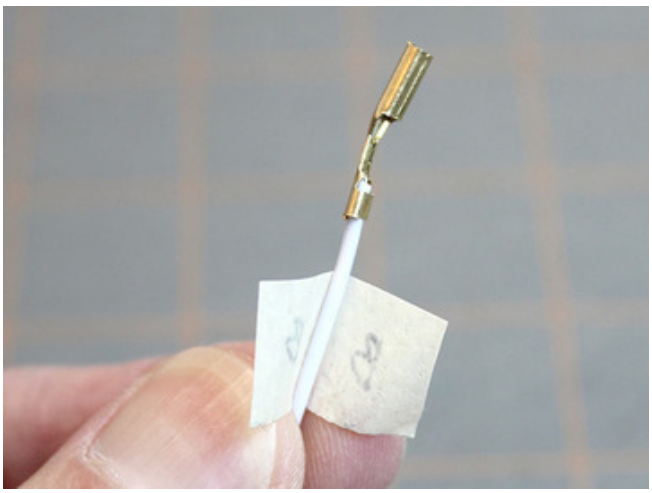
How you create these labels is up to you, depending on the materials and tools you have on hand. Ours are low-tech…masking tape and a Sharpie pen…but if you have a spiffy label maker, go to town.

Make 12 labels, something akin to this list:

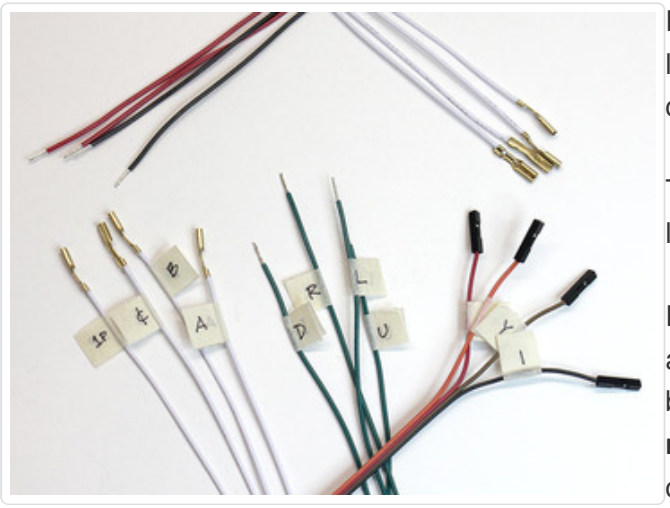
U (up)
D (down)
L (left)
R (right)
A (primary button)
B (secondary button)
X (joystick horizontal)
Y (joystick vertical)
+ (5V)
– (Ground)
¢ (Coin insert)
1P (1 Player Start)

Each label has two symbols written on it, so it's readable from either side.

X and **+** look similar enough that you may want to underline them for directional clarity: **X** and **+**.



The labels wrap around individual wires, near the connector end. But not yet…don't just wrap willy-nilly…these each need to go on specific wires!



Four of the white **quick-connect** wires should be labeled with: **A**, **B**, **¢** and **1P**. The other four quick-connect wires **do not receive labels**.

The four **conjoined jumper** wires should be labeled with: **X**, **Y**, **+** and **–**.

Four of the **plain** wires should be labeled: **U**, **D**, **L** and **R**. If following our color scheme, these would be the green wires. The other four plain wires **do not receive labels** (unless you're using a single color of wire, in which case we'd suggest adding extra **+** and **–** labels to two wires each.)

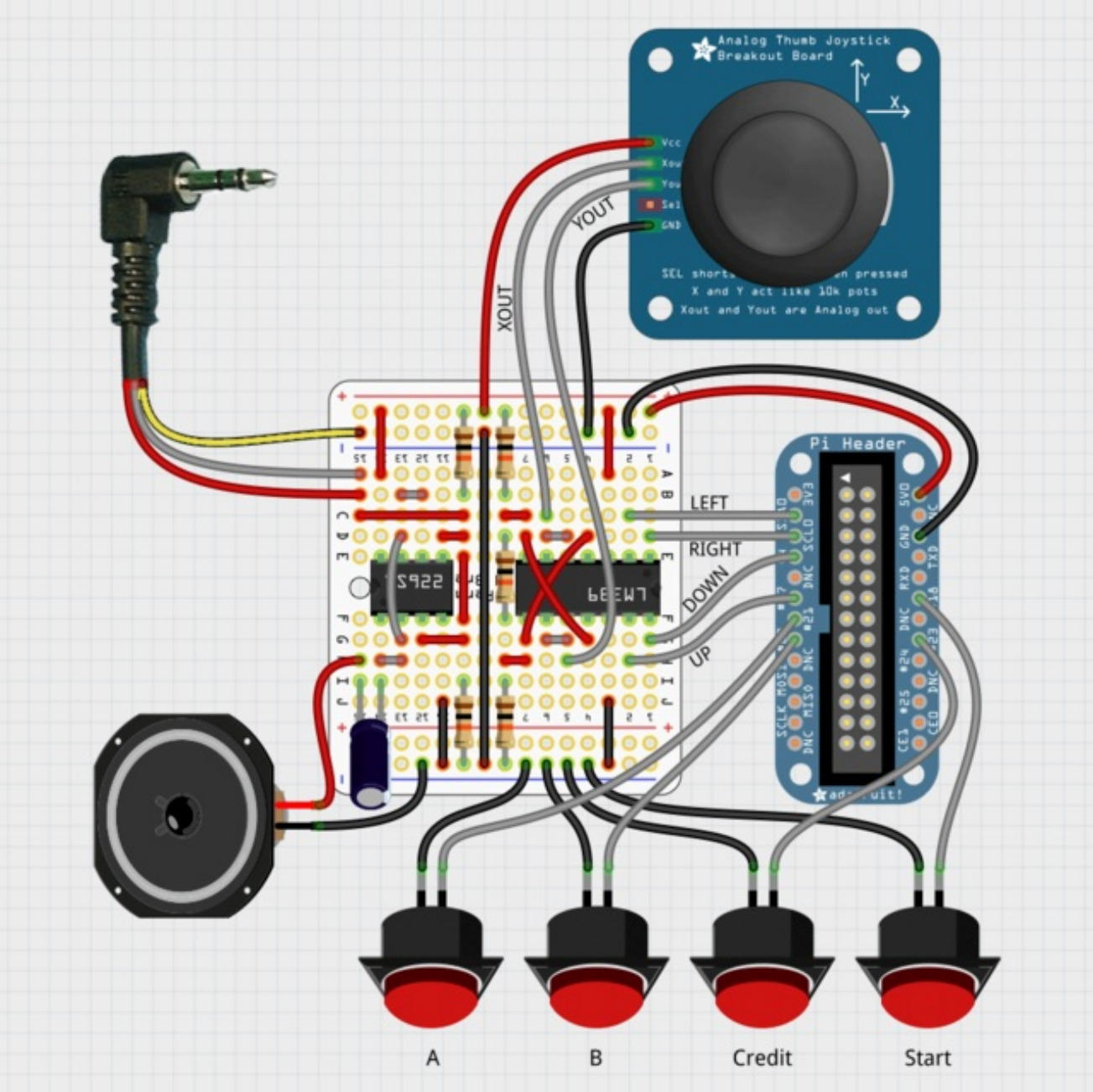
Cut some short pieces of heat-shrink tubing. Ten should suffice, about 1/4" long. Keep a little in reserve for a later step.

These will be used to hold wire bundles together, preventing breakage.

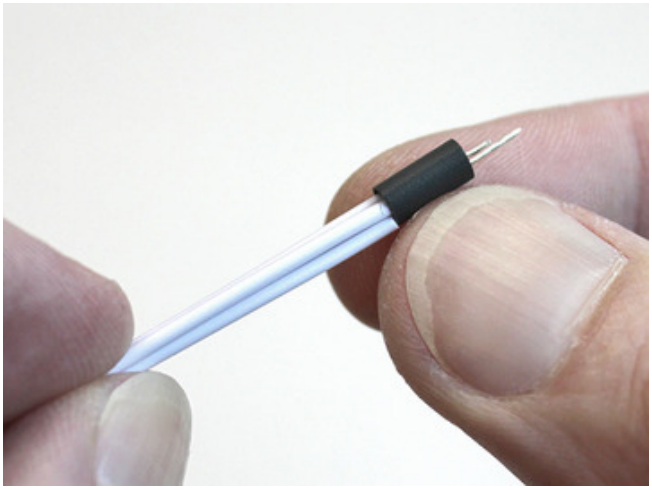
Beta Board Part 3

These directions apply only to the BETA Cupcade units. Most users can refer to the simplified final design on the “Interface Board” page.

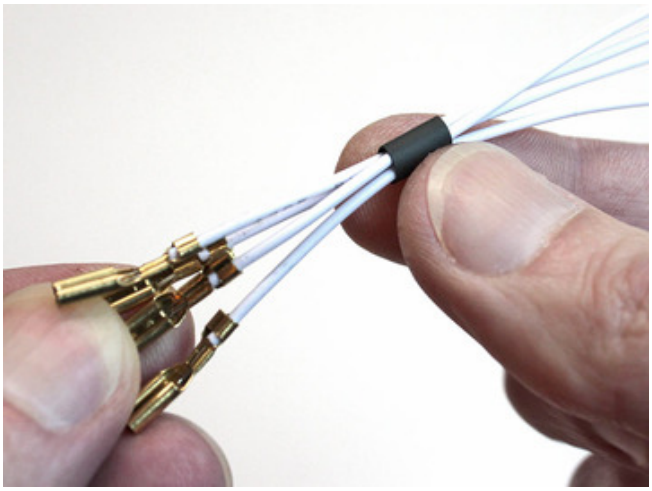
Let’s refer to this diagram again. The Perma-Proto circuit is done, now we’re placing all the long wires.



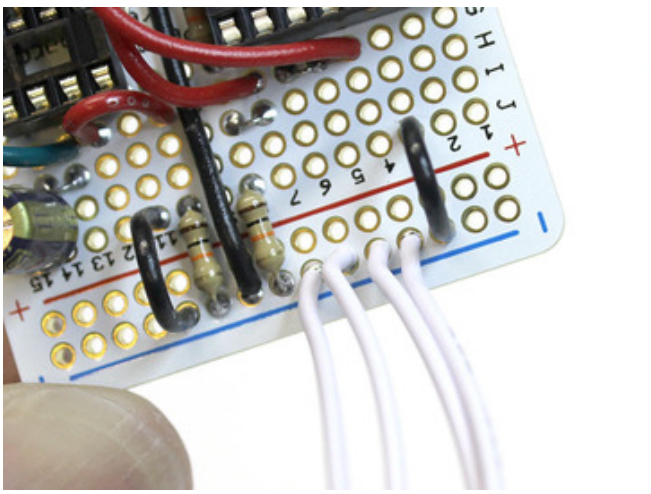
Button Wires



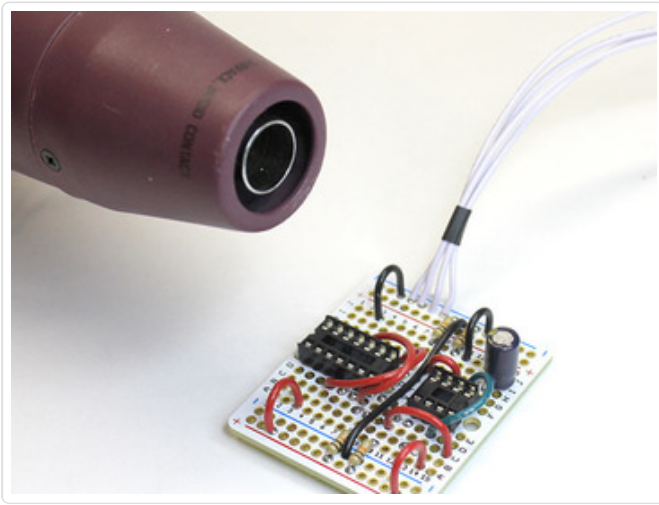
Gather up four quick-connects (the unlabeled ones) and squeeze the tips through a piece of heat-shrink tube.



Slide the heat-shrink all the way down, near the metal spades, but don't shrink it yet!

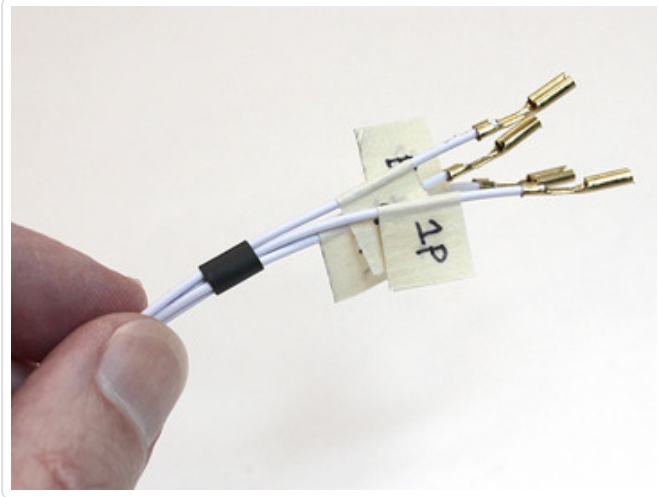


Solder the other end of these four wires to the lower ground bus on the Perma Proto board. The wires enter from the top and are soldered on the underside. Clip any excess wire protruding from the bottom.

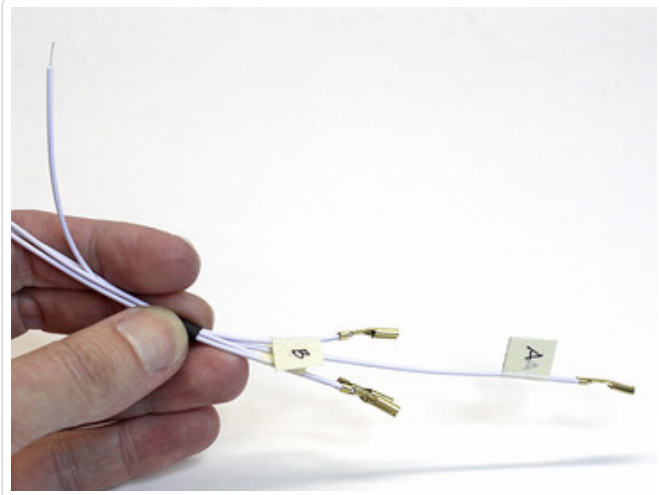


Now slide the heat-shrink back down the wires, about 3/4" from the board, and apply heat. Most people simply use a butane lighter for this, but if you have a heat gun that's even better.

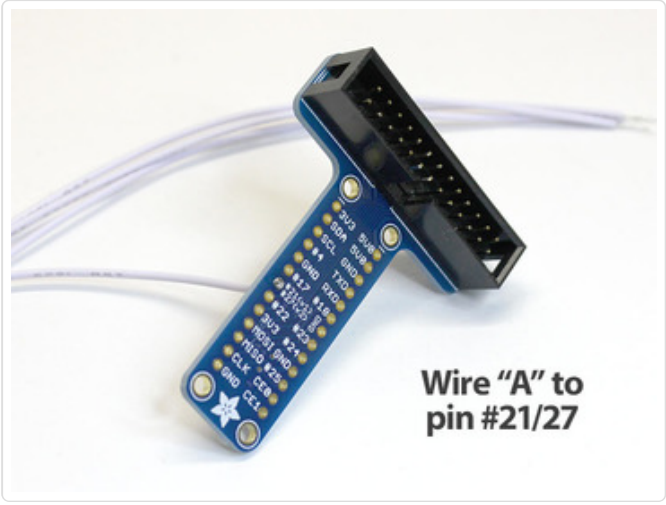
The heat-shrink tube lets these wires rely on each other for support; they're less likely to break off the board when cramming this into the case later.



Now do a similar thing with the four *labeled* quick-connect wires. Slide a piece of heat-shrink as far along the wires as it will go.

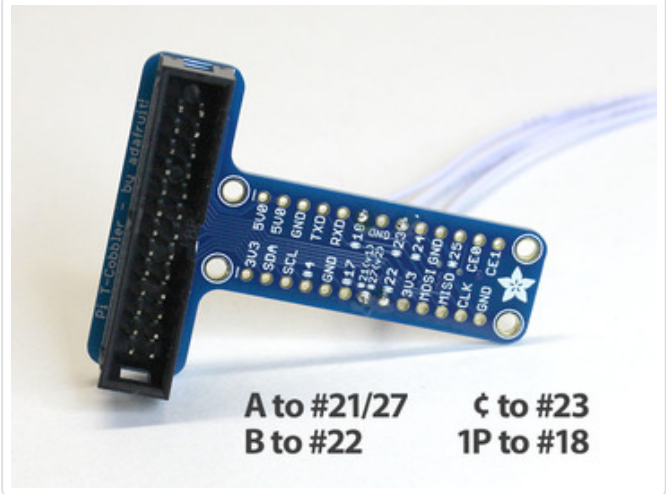


Now slide just <u>one wire</u> part way out, so we can tell it apart from the others at the bare end. Here we pulled the **A** wire and can identify its tail from the others.



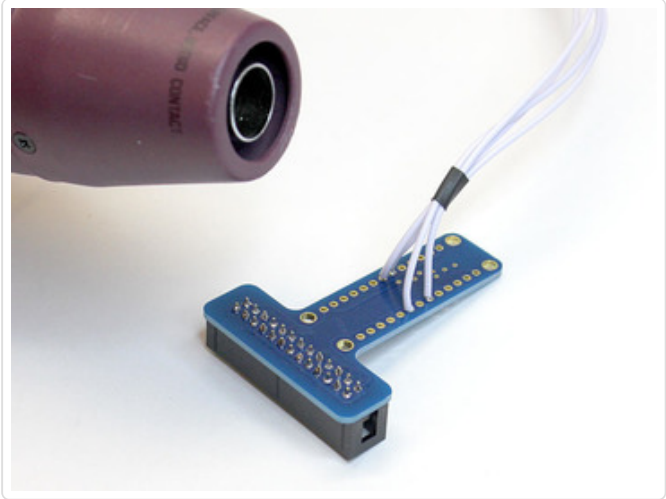
Using the schematic at the top of this page for reference, solder the other end of the wire to the appropriate point on the T-Cobbler. Wire **A** goes to Pin **#21/27**

The wires should enter from **below** the Cobbler and are **soldered on the top**. This is the opposite of the Perma Proto circuit. Clip away any extra protruding wire after soldering.



Repeat with the other three wires, one at a time so you don’t get them mixed up. Slide one wire part way out of the heat-shrink, grab the “tail” and solder in to the correct spot:

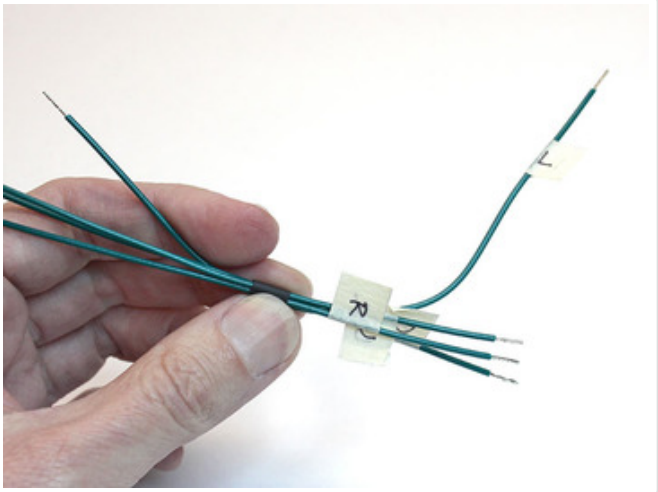
A to **#21/27**
B to **#22**
¢ to **#23**
1P to **#18**



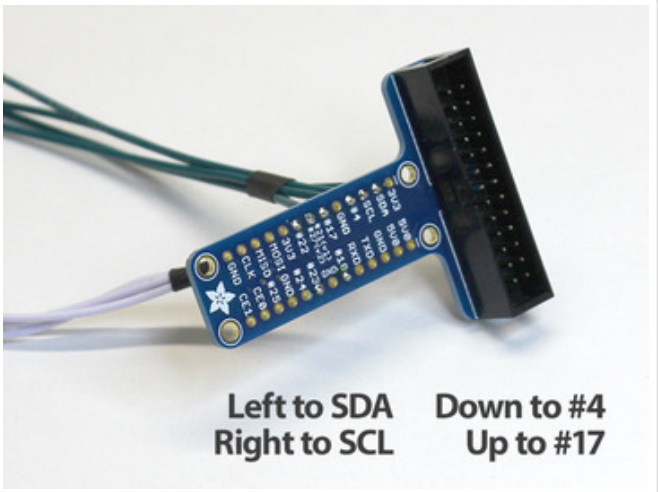
Slide the heat-shrink tube down close to the Cobbler and apply heat.

It's so *tidy!*

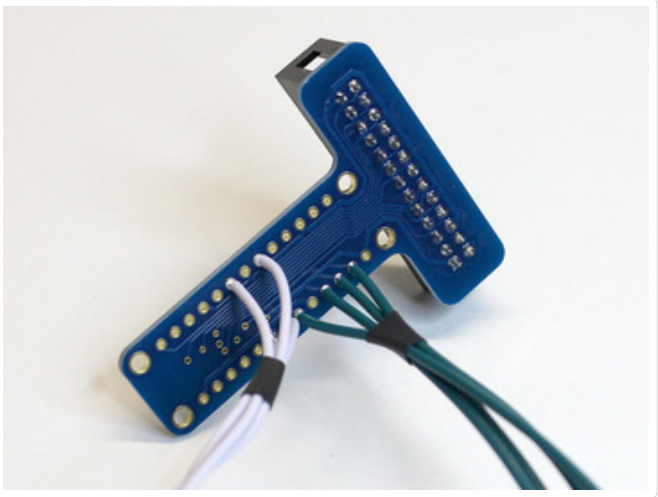
Joystick Wires



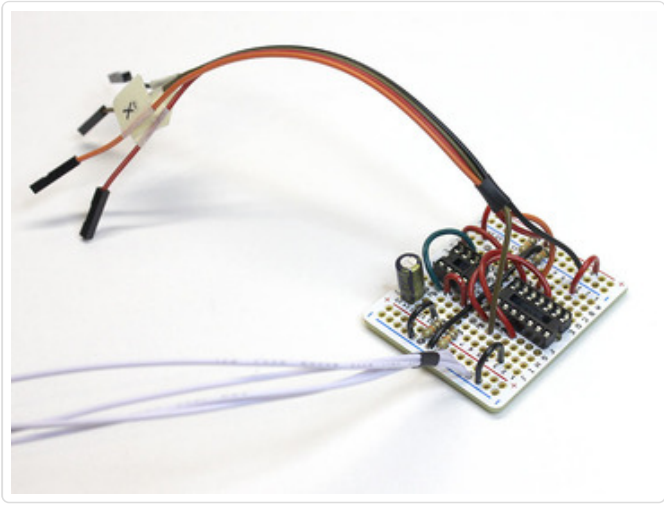
It's a very similar game with the U/D/L/R wires: squeeze them all through a piece of heat-shrink tube, then pull out one wire at a time, soldering the tail (unlabeled) end to the appropriate locations on the Cobbler…



L to **SDA**
R to **SCL**
D to **#4**
U to **#17**



Scoot the heat-shrink down close to the Cobbler and…you know the drill.

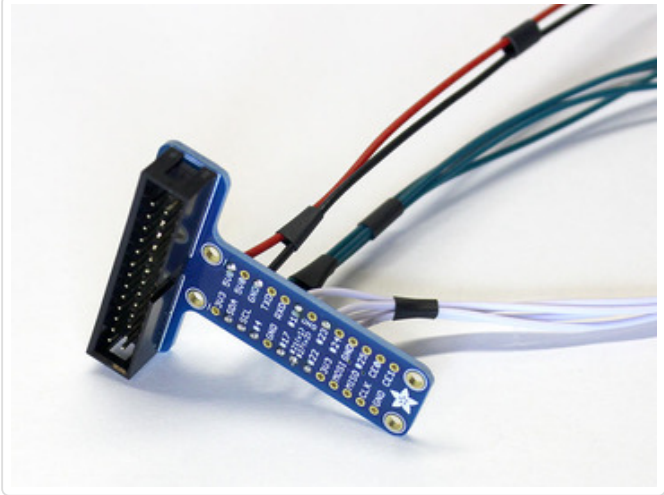


Returning to the Perma-Proto for a moment…

Solder the four ribbon jumpers (**+**, **X**, **Y**, **–**) to the appropriate points on the Perma-Proto circuit, using the diagram at the top of this page for reference.

Since these wires are conjoined, they don't really *need* heat-shrink tube, but you can add a piece if it makes you feel better. Remember to slide this on *before* any soldering.

Power Wires

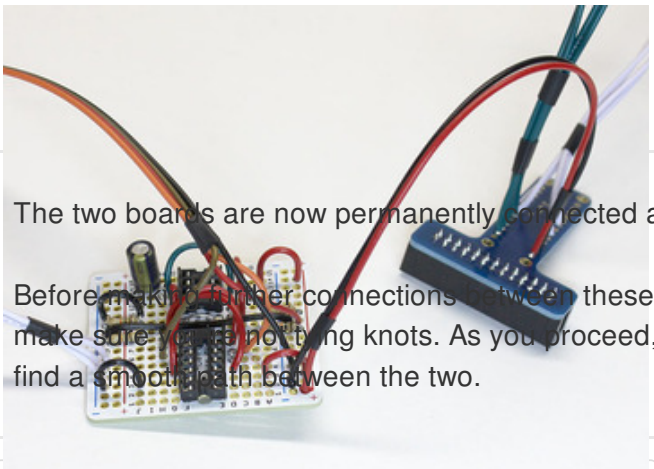


Add plain wires to the **+5V** and **GND** pins on the Cobbler (color-coded if you have them…if not, label these wires to avoid trouble).

These wires get *two* heat-shrink bits.

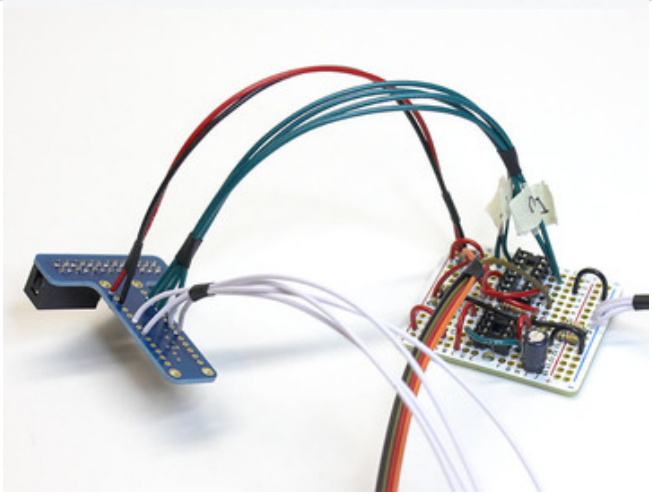
Notice also a second piece of heat-shrink has been added to the green joystick wires. But don't shrink any of these yet.

Solder the other end of the power wires to the appropriate positions on the Perma-Proto board, then heat-shrink the tubes near the ends.



The two boards are now permanently connected and will conspire to make your life miserable.

Before making further connections between these two boards or any other parts, plan it out first… make sure you're not tying knots. As you proceed, turn the boards different ways as necessary to find a smooth path between the two.

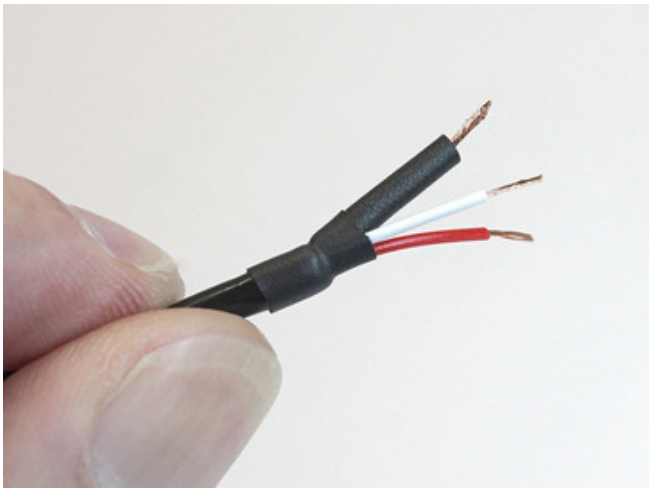


The four direction wires (**U**, **D**, **L**, **R**) can now be soldered to the appropriate positions on the Perma-Proto board. Refer to the schematic.

Now you can shrink that second piece of tube on the joystick wires.

See how the two wire bundles aren't fighting each other or making pretzels? That kind of order is what you're aiming for.

Audio Wires

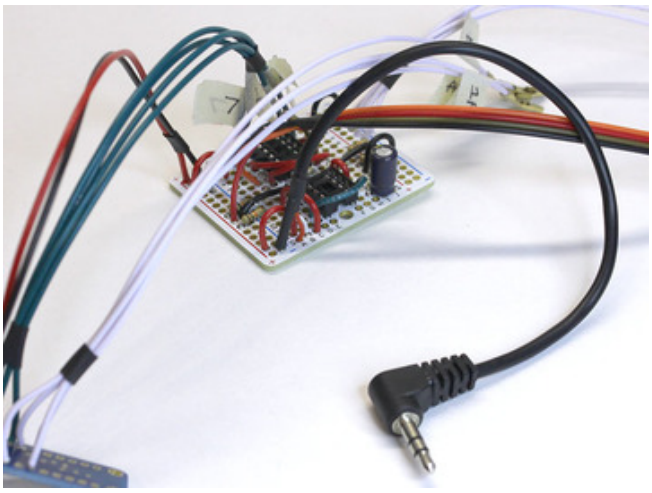


That bare copper wire on the audio cable is a problem…it connects to a ground point *right next* to a +5V line. **We must insulate!**

Slide a piece of heat-shrink tube a few inches down the cable (surrounding the whole thing, not just the copper wire).

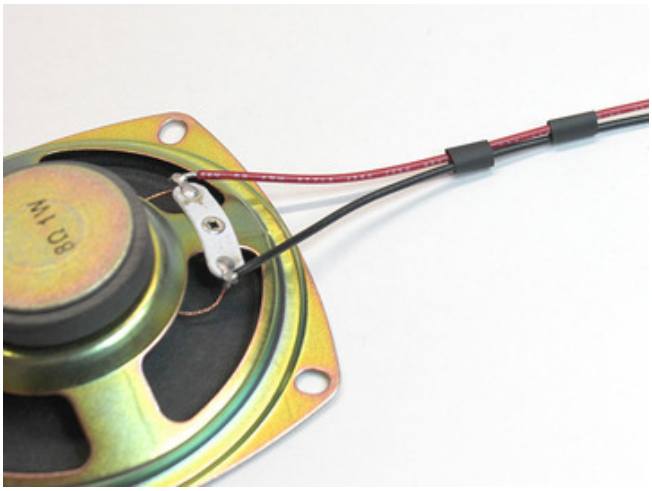
Slide a second piece of heat-shrink tube over just the copper wire (cut one a little longer from your remaining unused tube) and heat it up. This covers most of the wire, but there's still a tiny gap.

Slide the first piece back up so it covers the gap, apply heat.



Solder the three wires to the proscribed locations on the Perma-Proto circuit.

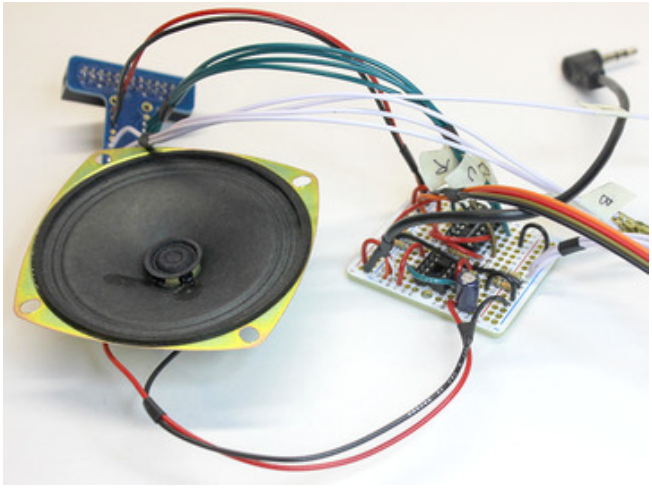
If you have a headphone cable with red and black and copper wires, treat the black wire as if it were the white wire. The 'raw' copper wire is always the ground wire!



Solder two plain wires to the connection points on the speaker.

You can color-code these if you like, but it's really not vital.

Then add two pieces of heat-shrink tube. Don't shrink yet!



Solder the opposite end of the wires to the corresponding points on the amplifier circuit, then shrink the tubing near each end.

Soldering's done!

The chips can now be inserted in the sockets. They need to be installed "back to back" — pin 1 (the end of the chip with the little bite missing) is in a **different orientation** for each: